Deep Learning-Based Spatio-Temporal Data Mining Using Multi-Source Geospatial Data

Bingnan Li

A thesis in fulfilment of the requirements for the degree of

Doctor of Philosophy

School of Civil and Environmental Engineering

Faculty of Engineeringerefefeff

The University of New South Wales

December 2022

THE UNIVERSITY OF NEW SOUTH WALES
Thesis/Dissertation Sheet

Surname or Family name: **Li**

First name: **Bingnan** Other name/s:

Abbreviation for degree as given in the University calendar: **PhD**

School: **School of Civil and Environmental Engineering** Faculty: **Faculty of Engineering**

Title: Deep Learning-Based Spatio-Temporal Data Mining Using Multi-Source Geospatial Data

Abstract

With the rapid development of various geospatial technologies including remote sensing, mobile devices, and Global Position System (GPS), spatio-temporal data are abundantly available nowadays. Extracting valuable knowledge from spatio-temporal data is of crucial importance for many real-world applications such as intelligent transportation, social services, and intelligent distribution. With the fast increase of the amount and resolution of spatio-temporal data, traditional data mining methods are becoming obsolete. In recent years, deep learning models such as Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) have made promising achievements in many fields based on the strong ability in automated feature extraction and have been broadly used in different spatio-temporal data mining tasks. Many methods have been developed, and more diverse data were collected in recent decades, however, the existing methods have faced challenges from multi-source geospatial data. This thesis investigates four efficient techniques in different scenarios for spatio-temporal data mining that take advantage of multi-source geospatial data to overcome the limitations of traditional data mining methods.

This study investigates spatio-temporal data mining from four different perspectives. Firstly, a multi-elemental geolocation inference method is proposed to predict the location of tweets without geo-tags. Secondly, an optimization model is proposed to detect multiple Areas-of-Interest (AOIs) simultaneously and solve the multi-AOIs detection problem. Thirdly, a multi-task Res-U-Net model with attention mechanism is developed for the extraction of the building roofs and the whole building shapes from remote sensing images, then an offset vector method is used to detect the footprints of the high-rise buildings based on the boundaries of the corresponding building roofs and shapes. Lastly, a novel decoder fusion model is introduced to extract interior road network from remote sensing images and GPS trajectory data. And this method is effective for multi-source data mining.

The proposed four methods use different techniques for spatio-temporal data mining to improve the detection performance. Numerous experiments show that the techniques developed in this thesis can detect ground features efficiently and effectively and overcome the limitations of conventional algorithms. The studies demonstrate that exploiting spatial information from multi-source geospatial data can improve the detection accuracy in comparison with single-source geospatial data.

ORIGINALITY STATEMENT

☑ I hereby declare that this submission is my own work and to the best of my knowledge it contains no materials previously published or written by another person, or substantial proportions of material which have been accepted for the award of any other degree or diploma at UNSW or any other educational institution, except where due acknowledgement is made in the thesis. Any contribution made to the research by others, with whom I have worked at UNSW or elsewhere, is explicitly acknowledged in the thesis. I also declare that the intellectual content of this thesis is the product of my own work, except to the extent that assistance from others in the project's design and conception or in style, presentation and linguistic expression is acknowledged.

COPYRIGHT STATEMENT

☑ I hereby grant the University of New South Wales or its agents a non-exclusive licence to archive and to make available (including to members of the public) my thesis or dissertation in whole or part in the University libraries in all forms of media, now or here after known. I acknowledge that I retain all intellectual property rights which subsist in my thesis or dissertation, such as copyright and patent rights, subject to applicable law. I also retain the right to use all or part of my thesis or dissertation in future works (such as articles or books).

For any substantial portions of copyright material used in this thesis, written permission for use has been obtained, or the copyright material is removed from the final public version of the thesis.

AUTHENTICITY STATEMENT

☑ I certify that the Library deposit digital copy is a direct equivalent of the final officially approved version of my thesis.

UNSW is supportive of candidates publishing their research results during their candidature as detailed in the UNSW Thesis Examination Procedure.

Publications can be used in your thesis in lieu of a Chapter provided:

- You contributed **greater than 50%** of the content in the publication and are the "primary author", i.e. you were responsible primarily for the planning, execution and preparation of the work for publication.
- You have approval to include the publication in their thesis in lieu of a Chapter from your Supervisor and Postgraduate Coordinator.
- The publication is not subject to any obligations or contractual agreements with a third party that would constrain its inclusion in the thesis.

☑ **Some of the work described in my thesis has been published and it has been documented in the relevant Chapters with acknowledgement.**

A short statement on where this work appears in the thesis and how this work is acknowledged within chapter/s:

Parts of Chapter 3 have been published in Communications in Li, B., Chen, Z., and Lim, S. (2021). Geolocation Inference Using Twitter Data: A Case Study of COVID-19 in the Contiguous United States. Communications in Computer and Information Science, vol 1411. Springer, Cham. https://doi.org/10.1007/978-3-030-76374-9_8.
Parts of Chapter 4 have been published in Li, B., Chen, L., Xiong, D., Chen, S., He, R., Sun, Z., Lim, S., and Jiang, H. (2022). Simultaneous Detection of Multiple Areas-of-Interest Using Geospatial Data from an Online Food Delivery Platform. ACM SIGSPATIAL '22: Proceedings of the 30th International Conference on Advances in Geographic Information Systems, pp. 1-10. https://doi.org/10.1145/3557915.3561014.
Parts of Chapter 5 have been published in Li, B., Gao, J., Chen, S., Lim, S., and Jiang, H. (2022). POI Detection of High-Rise Buildings Using Remote Sensing Images: A Semantic Segmentation Method Based on Multi-Task Attention ResU-Net. IEEE Transactions on Geoscience and Remote Sensing, vol 60, pp. 1-16. https://doi.org/10.1109/TGRS.2022.3174399.
Parts of Chapter 6 have been submitted in Li, B., Gao, J., Chen, S., Lim, S., and Jiang, H. (2022). Interior Road Extraction within Residential Complexes: A Decoder Fusion Model Leveraging Remote Sensing Images and GPS Trajectories. IEEE Transactions on Geoscience and Remote Sensing.

CANDIDATE'S DECLARATION

☑ I declare that I have complied with the Thesis Examination Procedure.

Abstract

With the rapid development of various geospatial technologies including remote sensing, mobile devices, and Global Position System (GPS), spatio-temporal data are abundantly available nowadays. Extracting valuable knowledge from spatio-temporal data is of crucial importance for many real-world applications such as intelligent transportation, social services, and intelligent distribution. With the fast increase of the amount and resolution of spatio-temporal data, traditional data mining methods are becoming obsolete. In recent years, deep learning models such as Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) have made promising achievements in many fields based on the strong ability in automated feature extraction and have been broadly used in different spatio-temporal data mining tasks. Many methods have been developed, and more diverse data were collected in recent decades, however, the existing methods have faced challenges from multi-source geospatial data. This thesis investigates four efficient techniques in different scenarios for spatio-temporal data mining that take advantage of multi-source geospatial data to overcome the limitations of traditional data mining methods.

Firstly, we developed a method of geolocation inference based on the whole potential location-related metadata of tweets. A crude form of geographic coordinate information can be obtained from every tweet's bounding box, while location-related information can be mined from the textual content, user location, and place labels via Named Entity Recognition (NER) techniques. Three coordinate datasets of the

United States counties were built and used as the coordinate references.

Secondly, a novel approach is proposed to detect multiple Areas-of-Interest (AOIs) simultaneously and solve the multi-AOIs detection problem. In this approach, we first applied the existing single-AOI detection algorithms to generate candidate spatial boundaries for AOIs in a neighborhood, and then developed a Binary Integer Linear Programming (BILP) model to determine the best candidate spatial boundaries for these AOIs while accounting for their spatial dependency.

Thirdly, a multi-task Res-U-Net model with attention mechanism is developed for the extraction of the building roofs and the whole building shapes from remote sensing images, then an offset vector method is used to detect the footprints of the high-rise buildings based on the boundaries of the corresponding building roofs and shapes. The OFD data were also applied to parse the POI name of every building footprint.

Fourthly, we developed a decoder fusion model based on the dilated Res-U-Net (DF-DRUNet) which fuses the remote sensing images and GPS trajectories in a more efficient way to extract the interior road network. The DF-DRUNet model is built on two components: Firstly, two independent dilated Res-U-Net models with each taking remote sensing images and GPS trajectories as input modalities respectively. Then, we fused the decoders from two modalities based on a dual fusion module, which can help to learn the modality selection from these two modalities.

The proposed four methods use different techniques for spatio-temporal data mining to improve the detection performance. This study demonstrates that exploiting spatial information from multi-source geospatial data can improve the detection accuracy in comparison with single-source geospatial data. Our experiments show that these techniques developed in this thesis can detect ground features efficiently and effectively, and overcome the limitations of conventional algorithms.

Acknowledgement

In the journey of my PhD study, I received precious guidance, support, and encouragement from a number of people. First of all, I would like to express my profound gratitude to my primary supervisor, Prof. Samsung Lim for his advice, guidance, encouragement and consistent support since the first day I came to UNSW. I am also deeply grateful to my secondary supervisors, Prof. Hai Jiang from Tsinghua University, and Dr. Xin Cao from the School of Computer Science and Engineering, UNSW for their continuous support and invaluable comments from time to time. This study would not be completed without their support. I also want to thank the anonymous reviewers for their careful reading and helpful comments which significantly helped in improving the journal papers.

I would like to convey my appreciation to Mrs. Warassamon Kate Brown, Mrs. Denise Lee, and Mrs. Ellie Williams for providing me with work opportunities and to other staff members of the School of Civil and Environmental Engineering for their support and devotion to ensure the workplaces are enjoyable and nice.

I would like to thank my numerous colleagues in the Surveying and Geospatial Engineering (SAGE) Research Group. In particular, Zi Chen, Qishuo Gao, Jiyu Liu, Yu Sun, Ziqi Ma, Kai Chen, Badal Pokharel, Yanzhi Wang, Gaochao Lin, Sharareh Akbarian, Maryamsadat Hosseini, Chang Liu, and Daniel Fowler, who have been helpful to me.

Due to the outbreak of the COVID-19 pandemic, I was stuck in China from January 2020. Luckily, Prof. Hai Jiang recommended me to participate in a research project between Tsinghua University and Meituan, and I had a research internship in the Map Algorithm Strategy Group of Meituan. I would like to express my gratitude to my colleagues in Meituan, including Jiuchong Gao, Shuiping Chen, Daping Xiong, Liying Chen, Jiawei Li, Jie Zhao, Yatong Song, Jia Shi, Ai Ji, Linkun Lv, Shuai Li, Junjie Xu, and Yuan Zhou.

I would like to appreciate the financial support from China Scholarship Council (CSC) and UNSW. Thanks for providing me the living stipends and tuition fee scholarship during my research. Without your support it is impossible for me to accomplish the journey.

Finally, I would like to express my deepest gratitude to my parents, my partner, my sister, my brother-in-law, and my nephew for their boundless love, support, and encouragement throughout my PhD study.

List of Publications

The publications which have been published in journals and presented in the conference proceedings during the period of my PhD study are list as follows.

1. **Li, B.**, Gao, J., Chen, S., Lim, S., and Jiang, H. (2022). POI Detection of High-Rise Buildings Using Remote Sensing Images: A Semantic Segmentation Method Based on Multi-Task Attention Res-U-Net. *IEEE Transactions on Geoscience and Remote Sensing*, vol 60, pp. 1-16. https://doi.org/10.1109/TGRS.2022.3174399.

2. **Li, B.**, Chen, L., Xiong, D., Chen, S., He, R., Sun, Z., Lim, S., and Jiang, H. (2022). Simultaneous Detection of Multiple Areas-of-Interest Using Geospatial Data from an Online Food Delivery Platform. *ACM SIGSPATIAL '22: Proceedings of the 30th International Conference on Advances in Geographic Information Systems*, pp. 1-10. https://doi.org/10.1145/3557915.3561014

3. **Li, B.**, Chen, Z., and Lim, S. (2021). Geolocation Inference Using Twitter Data: A Case Study of COVID-19 in the Contiguous United States. *Communications in Computer and Information Science*, vol 1411. Springer, Cham. https://doi.org/10.1007/978-3-030-76374-9_8.

4. **Li, B.**, Chen, Z., and Lim, S. (2020). Geolocation Prediction from Tweets: A Case Study of Influenza-like Illness in Australia. In *Proceedings of the 6th International Conference on Geographical Information Systems Theory, Applications and Management - GISTAM*, ISBN 978-989-758-425-1; ISSN 2184-500X, pp. 160-167. http://doi.org/10.5220/0009345101600167 .

5. **Li, B.**, Gao, J., Chen, S., Lim, S., and Jiang, H. (2022). Interior Road Extraction within Residential Complexes: A Decoder Fusion Model Leveraging Remote Sensing Images and GPS Trajectories. *IEEE Transactions on Geoscience and Remote Sensing*. (Submitted on 16 August 2022; Under Review)

6. Chen, Z., Pokharel, B., **Li, B.**, and Lim, S. (2021). Location Extraction from Twitter Messages Using a Bidirectional Long Short-Term Memory Neural Network with Conditional Random Field Model. *Communications in Computer and Information Science*, vol 1411. Springer, Cham. https://doi.org/10.1007/978-3-030-76374-9_2.

7. Chen, Z., Pokharel, B., **Li, B.**, and Lim, S. (2020). Location Extraction from Twitter Messages using Bidirectional Long Short-Term Memory Model. In *Proceedings of the 6th International Conference on Geographical Information Systems Theory, Applications and Management - GISTAM*, ISBN 978-989-758-425-1; ISSN 2184-500X, pp. 45-50. https://doi.org/10.5220/0009338800450050.

Contents

List of Figures

List of Tables

Abbreviations

AG	Attention Gate
AOI	Area-of-Interest
AOIs	Areas-of-Interest
APIs	Application Programming Interfaces
BCE	Binary Cross-Entropy
BILP	Binary Integer Linear Programming
BN	Batch Normalization
CNN	Convolutional Neural Network
DBA	Digital Boundary's Average
DBC	Digital Boundary's Centroid
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DFM	Dual Fusion Module
DL	Deep Learning
DNN	Deep Neural Network
DRUNet	Dilated U-Net with Residual Block
ESRI	Environmental Systems Research Institute
FCN	Fully Convolutional Network
FN	False Negative
FP	False Positive

GA	Gazetteers of America
GIS	Geographic Information Systems
GOIs	Geometries-of-Interest
GPS	Global Positioning System
GPU	Graphics Processing Unit
HMM	Hidden Markov Model
ID	Identification
IDF	Inverse Document Frequency
IoU	Intersection over Union
JSON	JavaScript Object Notation
KDE	Kernel Density Estimation
LiDAR	Light Detection and Ranging
MDED	Median Error Distance
MED	Mean Error Distance
ML	Machine Learning
NER	Named Entity Recognition
NLP	Natural Language Processing
NLTK	Natural Language Toolkit
OFD	Online Food Delivery
OSM	OpenStreetMap
PCC	Pearson's Correlation Coefficient
POI	Point-of-Interest
POIs	Points-of-Interest
POS	Part-of-Speech
ReLU	Rectified Linear Unit

ResNet	Residual Network
RF	Random Field
ROI	Region-of-Interest
RS	Remote Sensing
TF	Term Frequency
TF-IDF	Term Frequency-Inverse Document Frequency
TIN	Triangulated Irregular Network
TN	True Negative
TP	True Positive
UNSW	University of New South Wales
UPTB	User location, Place label, Textual content, and Bounding box
URL	Uniform Resource Locator
US	United States
VGI	Volunteered Geographic Information
VIA	VGG Image Annotator

Chapter 1

Introduction

1.1 Background

Nowadays, huge amounts of data are recorded from different sources, such as public transportation [1], social media [2], and online food delivery platforms [3]. However, with the rapid development of different techniques including remote sensing, mobile devices, and GPS, spatio-temporal data have become progressively available [4]. This sheer scale of spatio-temporal data provides a great opportunity for researchers to discover valuable knowledge [5].

Discovering valuable knowledge using spatio-temporal data is extremely important for some real-world applications, such as social services, smart transportation, urban planning, and intelligent distribution [6]. With the fast increase of the amount, capacity, and resolution of spatio-temporal data, traditional data mining methods, especially those based on statistics, are becoming increasingly difficult to deal with those data [4]. In recent years, deep learning models such as CNN and RNN have made remarkable achievements in many fields due to the powerful automatic learning

and feature representation capabilities [5]. These models are also widely used in various Spatio-Temporal Data Mining (STDM) applications such as self-driving cars [7], ground targets classification [8], epidemic disease forecasting [9], and traffic management [10].

The rapid development of deep learning can be attributed to three major developments: a large amount of data, high-performance computing, and algorithm improvement. (1) Data: With the growth of the Internet, huge amounts of data are being generated by various sensors every day. (2) Computing: With the development of cloud computing and Graphics Processing Units (GPUs), the computing power of machines is becoming very powerful. (3) Algorithms: Researchers have proposed some of the most efficient methods of training deep learning models by algorithm improvements [11].

In this thesis, four spatio-temporal data mining techniques are investigated from different perspectives. Firstly, a multi-elemental geolocation inference method is proposed to predict the location of tweets without geo-tags. Secondly, an optimization model is proposed to detect multi-AOIs simultaneously and solve the multi-AOIs detection problem. Thirdly, a multi-task Res-U-Net model with an attention mechanism and an offset vector method are developed for the extraction of the building footprints. Lastly, a novel decoder fusion model is introduced to extract interior road network from remote sensing images and GPS trajectory data. These four data mining techniques can improve the feature detection performance of advanced geospatial applications. This study demonstrates that exploiting spatial information from multi-source geospatial data can improve the detection accuracy in comparison with single-source geospatial data. Our experiments show that these four techniques can detect ground features efficiently and effectively, and overcome the limitations of conventional algorithms.

1.2 Motivations

In this section, we present the motivations of four research topics that are useful in many spatio-temporal data mining applications such as geolocation inference and ground targets extraction.

1.2.1 Geolocation Inference Using Twitter Data

Over the past decade, the Internet has helped revolutionize every aspect of people's lives, and it is not only a source to get information, but also a platform to disseminate personal information [12, 13]. In addition, the development of mobile devices makes it easier to send digital information. At the same time, social media platforms have experienced a tremendous and profound reform. Twitter and Facebook mainly provide basic services, but other types of social media are being used to connect online for different reasons, such as location-based services, media sharing services, as well as other types of services. Users can establish online friendships based on mutual interests and share their everyday life with each other.

Supported by previous studies [14–16], Twitter outshines other platforms in regard to social network analysis and event detection because of not only its excellent design, but also its vast user base of different age groups. The large quantity of user-generated contents is employed for data mining in various research areas [12]. Tweets with accurate geographic information can provide significant benefits to event response and monitoring, hence those without geographic information become useless unless geolocation inference is applicable. Accurate prediction of tweets' geolocation can effectively benefit the response and rescue in emergency events [17].

The development of GPS-enabled mobile devices enables users to share and track their locations with accurate geographical coordinates. However, due to the opera-

tional complexity and privacy concerns, most users do not turn this function on [18]. As Laylavi *et al.* [19] illustrated, the percentage of tweets with geo-tags accounts for only 2%, which severely limits the development of associated applications. Therefore, accurate geolocation inference of tweets has become an urgent problem in this research field.

Nowadays, disease-related information is increasingly shared in real time through Twitter, while timely data with spatial and temporal information play a significant role in surveillance of an epidemic disease [20, 21]. Every single tweet has its own metadata, which includes its creation time, but under most circumstances, does not contain its created geographical coordinates, hence geolocation inference of tweets is still a critical issue. Real-time data without any geographic information can be almost meaningless for emergency response and surveillance of an epidemic disease. Thus, this study aims to develop novel methods to predict the geolocation of non-geotagged tweets based on their own metadata.

In this study, models based on multiple attributes of the tweet's metadata are built to predict the non-geotagged tweets' geolocation. Attributes of textual content, user location, place labels, and bounding box are fully used during the modelling process. The dataset used in this study was collected between the 10th and 30th of June 2020. During this time, the United States (US) was suffering a severe effect of the COVID-19 pandemic. The development of technologies, including Natural Language Processing (NLP) and Named Entity Recognition (NER) make it easier to extract location entities from textual data.

1.2.2 Detection of Multi-AOIs by OFD Data

With the development of mobile Internet, OFD services have become increasingly popular in our daily lives [22]. According to Stock Apps, in 2020, the global OFD

market reached $136.4 billion in revenue, an increase of 27% year-over-year, and the number of users reached 1.46 billion, a 25% increase from 2019 [23]. China is leading the way in terms of market size and has achieved great success [24]. Take Meituan, one of the world's largest food delivery platforms as an example. In 2020, total revenues reached 114.8 billion yuan, an increase of about 18% year-over-year, and the number of users reached around 460 million, a 13% increase from 2018 [25].

OFD platforms rely heavily on accurate Areas-of-Interest (AOIs) information in their operations. An AOI, also known as a Region-of-Interest (ROI), refers to a polygon selection in a map that someone may find useful or interesting, for example, a residential complex, a public park or a shopping mall [26, 27]. OFD platforms are concerned with two key properties associated with an AOI, that is, its name and spatial boundary. Spatial boundaries of AOIs are stored as vector formats, which can be easily used in geospatial analysis and improve service efficiency of the OFD industry.

In this study, we investigate how to detect AOIs using the geospatial data collected from OFD platforms. The AOI detection problem involves identifying the name and the boundary of an underlying AOI. Although there has been a proliferation of studies that investigate the AOI detection problem, existing approaches all focus on the single-AOI detection problem, that is, they detect AOIs one at a time. In fact, some noisy GPS points locate in the wrong locations, therefore, they may construct spatial boundaries of different AOIs with overlaps. Among existing studies, multiple geospatial data sources are applied into the single-AOI detection problem, including social media data (e.g., geo-tagged Flicker photos [28–30] and geo-tagged tweets [31–33]), remote sensing data [33], and delivery data [3]. Since single AOI detection algorithms detect AOIs independently of each other, they tend to produce inconsistent results and cannot fully leverage GPS data available for the adjacent AOIs.

In this study, we aim to address the challenge faced by single-AOI detection models through simultaneously detecting multiple AOIs. In our approach, we first apply existing single-AOI detection algorithms to generate candidate spatial boundaries for AOIs in a neighborhood, we then develop a Binary Integer Linear Programming (BILP) model to determine the best candidate spatial boundaries for these AOIs while accounting for their spatial dependency. We conducted numerical experiments using real data from Meituan, the largest OFD platform in China. Results show that our model not only produces consistent AOI boundaries but also improves the average F1-score. We improve the accuracy and preserve the details of the boundaries of detected AOIs by applying a Hidden Markov Model (HMM) to the road network dataset.

1.2.3 POI Detection Using Remote Sensing Images

OFD platforms rely heavily on accurate Points-of-Interest (POIs) information in their operations. A POI is defined as a specific point location that may be useful or interesting for people [31], e.g., a residential building footprint. OFD platforms are concerned with two key properties associated with a POI, that is, its name and spatial boundary. ROI mining techniques are designed to detect boundaries of POIs. Generally, the boundary of a POI is defined as a cluster, which is given by a geographically shaped convex polygon using the convex hull of the geotagged records that include a given textual description such as a POI name within a circle to be optimized [2]. These existing methods highly rely on the accurate GPS coordinates, while the accuracy of indoor GPS satellite signals is relatively low, and high-rise buildings can also block or degrade GPS satellite signals of mobile devices, which may cause users to locate wrong locations. Therefore, these methods cannot extract POI boundaries accurately.

As remote sensing technologies mature, the capability to detect the physical boundaries of ground objects has been enhanced significantly. Thus, remote sensing technologies are seen as conventional and even important methods for detecting building footprints [34], which are also regarded as boundaries of POIs. In this study, we extract spatial boundaries of POIs based on remote sensing images. For example, different buildings are located in a residential complex, and every one of them can be recognized as a specific POI, and we aim to identify the spatial boundary and the name of each specific POI based on buildings from remote sensing images and OFD data.

For most existing studies, remote sensing data focus on low-rise buildings, and therefore building footprints can be generally extracted from the "footprint" of the roof [35–37]. However, for high-rise buildings, due to the different view angles of remote sensing sensors, the distance between the polygons of the building roof and the building footprint on remote sensing images can be very large and changes with time. In this case, the polygon of the building roof cannot be regarded as the polygon of the building footprint. Besides building footprints cannot be extracted directly, because most of them are covered by building roofs and building surfaces on remote sensing images. Based on data analysis and calculation, we found that, in most cases, the building footprint has the same shape as the building roof, except for a certain offset in position.

In this study, we propose a multi-task Res-U-Net model with attention mechanism to extract the boundaries of building roofs and shapes simultaneously, and then use an offset vector method to detect the actual spatial boundaries of the building footprints based on the spatial boundaries of the building roofs and shapes. After detecting the spatial boundaries of the building footprints, parsing of POI names is processed by the OFD dataset. We conducted numerical experiments using remote sensing data from Google Earth and the OFD dataset from Meituan platform. Experimental

results indicate that the proposed method successfully extracts the boundaries of the building roofs and shapes, and achieves the best performance of F1-score and IoU for both the building roof segmentation and the whole building shape segmentation among baseline models.

1.2.4 Road Extraction Using Multi-Source Data

Roads within the residential complexes play a very important role in route planning of OFD services. If there is no interior road network within an AOI, the OFD platform can only direct the deliverers to the location of the gate area, and then the deliverers have to find the customer's place all by their familiarity with the place, which is inefficient and a waste of time. If the road network is completed for both the outside and the inside of the AOI, then a more efficient route to the customer's destination can be planned by the OFD platform.

In the past few years, a wide range of methods have been developed in the research area of road extraction from GPS trajectories. Although substantial progress has been achieved [38, 39], this research field still faces great challenges. For existing studies, a wide range of approaches have been proposed for road extraction using remote sensing images. Early studies mostly extracted hand-made features such as textures and contours, and used shallow models to identify road areas. In recent years, deep learning methods are becoming the mainstream of this research area because of their powerful automatic representation learning ability, and have made remarkable success. However, it still remains a very difficult issue to extract road regions using remote sensing images due to the issues of occlusion and similar appearance with other ground targets [1].

Due to the problem of information loss, it is still challenging to extract interior road regions based on a single data source. As remote sensing images and GPS

trajectories can provide different types of information, these can complement each other for interior road extraction. Remote sensing image-base approaches and GPS trajectory-based approaches have their own advantages and disadvantages. That is, the fusion of these two complementary data sources can provide an efficient way to take advantage of information for robust interior road extraction. However, the number of related studies [40, 41] that use the two modalities is very limited. In addition, most of these studies directly fuse the input layer with remote sensing images and GPS trajectory feature maps, which is not an ideal strategy for multi-modal fusion methods.

In this study, we propose a decoder fusion model named DF-DRUNet, which fuses remote sensing images and GPS trajectories in a more efficient way for extracting interior road network. The DF-DRUNet model is built on two components. First, two independent dilated Res-U-Net models with each taking remote sensing images and GPS trajectories as input modalities respectively. Second, we fuse decoders from two modalities based on a dual fusion module (DFM), which can help to learn the modality selection from these two modalities. Numerical experiments have been conducted based on the DF-DRUNet model and baseline models from the real dataset of remote sensing images and GPS trajectories.

1.3 Contributions

In this section, the main contributions of this thesis are summarized based on four spatio-temporal data mining tasks, which are described as follows.

(1) Geolocation Inference Using Twitter Data

- Exploring potential location-related attributes of the tweet's metadata and

extracting location entities via NER techniques.

- Three geographic coordinate datasets of counties are used to predict geolocation and the proposed models are built according to different priorities of location-related attributes.
- When the area threshold of the bounding box is set to 10,000 km^2, the best model can successfully predict the geolocation of 90.8% of COVID-19 related tweets with the mean error distance of 4.824 km and the median error distance of 3.233 km.
- The proposed method enhances the granularity of geographic information of tweets and makes the surveillance of COVID-19 effective and efficient.

(2) Simultaneous Detection of Multi-AOIs Using OFD Data

- By accounting for the spatial dependency among neighbouring AOIs, we ensured that our approach can produce AOI boundaries that are consistent with each other.
- We formulated the problem as a Binary Integer Linear Programming (BILP) model, which can be efficiently solved by standard branch-and-bound procedures.
- Using the optimization model in the dataset collected from Meituan platform, results show that our model identifies multi-AOIs and improves the average F1-score from 0.847 to 0.894 and achieves the best average F1-score among all single-AOI detection methods.

(3) POI Detection of High-Rise Buildings Using Remote Sensing Images

- We proposed a novel multi-task Res-U-Net model with attention mechanism for semantic segmentation of the building roofs and shapes. Using the proposed model, the building roofs and the whole building shapes are extracted simultaneously. Even compared with the best performing baseline model, the proposed model improves the total F1-score by 1.78% and IoU by 0.49% in terms of the building roof segmentation, and the total F1-score by 3.31% and IoU by 3.03% for the whole building shape segmentation.
- Most of existing studies extract building roofs as building footprints, while in our study, we introduced an offset vector method to extract the building footprints based on boundaries of the building roofs and the whole building shapes for high-rise buildings.
- Instead of detecting spatial boundaries of the building footprints, our research also parses POI names based on the OFD dataset.

(4) Interior Road Extraction Using Multi-Source Data

- We propose a novel decoder fusion model with the DFM unit based on two independent dilated Res-U-Net models for semantic segmentation of the interior road from both remote sensing images and GPS trajectories. The DF-DRUNet model achieves the best performance of F1-score and IoU among all baseline models.
- A novel DFM unit is designed to help to learn the modality selection from both the two modalities of remote sensing images and GPS trajectories.
- Most of existing studies extract road network outside residential complexes, while our study concentrates on interior road network extraction within residential complexes from multi-source data.

1.4 Thesis Structure

This thesis contains seven chapters. The main objective of each chapter is described below.

Chapter 1 presents an introduction of geolocation inference using twitter data, simultaneous detection of multiple AOIs using online food delivery data, POI detection of high-rise buildings using remote sensing images, and interior road extraction using multi-source data.

Chapter 2 describes a detailed literature review on the latest advances in geolocation inference using twitter data, simultaneous detection of multiple AOIs using online food delivery data, POI detection of high-rise buildings using remote sensing images, and interior road extraction using multi-source data.

Chapter 3 introduces a method of geolocation inference based on the whole potential location-related metadata of tweets. A crude form of geographic coordinate information can be obtained from every tweet's bounding box, while location-related information can be mined from the textual content, user location and place labels via NER techniques. Three coordinate datasets of the United States counties are built and used as the coordinate references. Models with different data sources have been employed to predict the geolocations of the tweets related to COVID-19 in the contiguous United States.

Chapter 4 proposes a new approach to detect multiple AOIs simultaneously and solves the multi-AOIs detection problem. In our approach, we first apply the existing single-AOI detection algorithms to generate candidate spatial boundaries for AOIs in a neighborhood, and then develop a BILP model to determine the best candidate spatial boundaries for these AOIs while accounting for their spatial dependency. We conduct numerical experiments using real data from Meituan, the largest OFD

platform in China.

Chapter 5 proposes a multi-task Res-U-Net model with attention mechanism for the extraction of the building roofs and the whole building shapes from remote sensing images, then use an offset vector method to detect the footprints of the high-rise buildings based on the boundaries of the corresponding building roofs and shapes. We also apply the OFD data to parse the POI name of every building footprint. Several strategies are also developed in combination with the proposed model, including data augmentation and post-processing. We conduct numerical experiments using real data of remote sensing images and OFD historical order data.

Chapter 6 develops a novel decoder fusion model based on dilated Res-U-Net which fuses the remote sensing images and GPS trajectories in a more efficient way to extract the interior road network. The DF-DRUNet model is built on two components: First, two independent dilated Res-U-Net models with each taking remote sensing images and GPS trajectories, respectively, are used as input modalities. Second, we fuse the decoders from two modalities based on a dual fusion module, which can help to learn the modality selection from these two modalities. Numerical experiments were conducted using the DF-DRUNet model and baseline models from the real dataset of remote sensing images and GPS trajectories.

Finally, Chapter 7 concludes the whole work of this thesis and suggests some potential directions for future research.

Chapter 2

Spatio-Temporal Data Mining Using Multi-Source Geospatial Data

Spatio-temporal data mining is playing a vital role in the big data era with increasing availability and importance of large spatio-temporal datasets such as social media data, remote sensing images and GPS trajectories. As mentioned in Chapter 1, there are also different types of methodological approaches for spatio-temporal data mining. As this research field has been developed rapidly over the past few years, this thesis attempts to systematically review the latest advances specifically with respect to our four research topics. In this chapter, we give an overview of the related work for spatio-temporal data mining problems studied in this thesis. We first describe the related work of geolocation inference using Twitter data in Section 2.1. Then the related work of simultaneous detection of multi-AOIs using online food delivery data are described in Section 2.2. In Section 2.3, we show the related work for POI detection of high-rise buildings using remote sensing images.

Finally, the related work of interior road extraction using multi-source data is illustrated in Section 2.4. These four research topics focus on the extractions of spatial objects, which are digital representations of the real world, but are exploited from four different perspectives. These four components constitute the whole research of spatio-temporal data mining using multi-source geospatial data.

2.1 Geolocation Inference Using Twitter Data

Social media applications are considered as an important part of people's everyday lives and people are more likely to move their interactions to the virtual platforms (e.g., Twitter, Instagram, Tiktok, and Facebook). As a result, social media applications have been considered to be one of the most effective and influential implications that have been gradually engaged in most aspects of individuals' lives [42]. Among these social media platforms, Twitter outshines others regarding social network analysis and event detection due to its vast user base of different age groups [14–16]. According to the most up-to-date Twitter statistics for 2022, its daily active users are around 211 million, which accounts for 23% of the Internet population, and about 500 million tweets are posted every single day [43].

Users sometimes add geo-information in their tweets, but in most cases, it is still neither complete nor accurate. Therefore, various methods and algorithms from other fields are being used in the field of geolocation inference. With the development of technologies such as machine learning, deep learning, NLP as well as Geographic Information Systems (GIS), much more methods have made breakthroughs in this research field [44]. However, different from formal articles which are well written and grammatically correct, social media messages always contain informal elements, e.g., acronyms, emojis, hashtags, and even typos, which is often attributed to the limit of character count and the use of mobile devices.

In the past few years, many studies of geolocation inference based on Twitter data have been published [44]. Ajao *et al.* [17] reviewed previous research related to geolocation inference of tweets, and summarized relevant methods and evaluation metrics. In the work of Cheng *et al.* [45], they discovered merely 20% of Twitter users in the US prefer to show cities where they live in their user profiles, and only 5% of them provide geographical coordinates information. The study of Hecht *et al.* [46] illustrated that even though self-described addresses are shown in their profiles, some of them are not accurate or valid, and geo-tagged tweets account for merely 0.77% of the whole. From the study of Ryoo *et al.* [47], the percentage of tweets with geographic information is only about 0.4%. Priedhorsky *et al.* [48] showed the similar percentages in their studies. More importantly, geolocation inference of social media data is the basis of other relevant studies. Consequently, further research in this area is needed.

When tweets are posted, some places information in the textual content enables us to understand them better. Textual content is used to predict the geolocation of tweets in the studies of Cheng *et al.* [49], Chandra *et al.* [50] as well as Chang *et al.* [51]. However, Ikawa *et al.* [52] described that some users always mention places that are not exactly where they are. In the study of Abrol *et al.* [53], they researched the social network relationships among their online friends. Backstrom *et al.* [54] and Bouillot *et al.* [55] described that geolocation inference of tweets can be achieved by the user profile in their studies.

NLP techniques enable various methods and algorithms of this field to be used in information extraction and geolocation inference. Techniques of NER and part-of-speech tagging (POS) have been introduced in the research of Lingad *et al.* [56]. Li *et al.* [57] introduced methods of machine learning and probabilistic to geolocation inference. Takhteyev *et al.* [58] used gazetteers and location databases in their research. In the study of Huang *et al.* [18], deep learning models are used to predict

geolocation of Twitter data. Previous studies have obtained a great achievement in this field and have the potential to pursue more accurate results of geolocation prediction [59].

Most studies conducted on geolocation inference of tweets focus on either textual content or other location-related attributes. The research gap of the above related studies is the geolocation inference using all the potential sources of location information of the Twitter dataset. And every data source can contribute to the geolocation inference. Therefore, this research aims to implement all feasible combinations of potential attributes related to location to predict the geolocation of tweets.

By following the literature review for data mining of Twitter dataset, data mining for a new data source of OFD data is introduced in the next chapter. Then the literature review of Area-Of-Interest (AOI) detection is described in detail.

2.2 Detection of Multi-AOIs Using OFD Data

With the development of mobile devices, volunteered geographic information (VGI) is another source of data for numerous spatial applications [60,61]. In addition, OFD information is a new type of geographic data generated by millions of customers and riders. Although the absence of quality control, some data have been demonstrated the same quality of the authoritative data [62]. Most of the existing studies are based on VGI data, such as geo-tagged Flicker photos, tweets, and other social media data [63–65].

Existing approaches for single-AOI detection can be broadly divided into three groups: pre-defined shapes, density-based clustering, and grid-based aggregation. And some other novel approaches are also introduced in this field.

Pre-defined shapes. In [29], the authors use circle with fixed radius to extract popular tourist routes based on geo-tagged Flickr photos. In particular, circles are used to represent a trajectory of coordinates into a series of AOIs. In [66], the authors use rectangular AOIs to represent stadiums in a study of trajectory pattern mining. Specifically, a stadium's AOI is the minimum rectangle of its area. Due to the lack of ability to construct complex polygons, this approach has significant limitations in practical application.

Density-based clustering. In [67], the authors apply the algorithm of Mean Shift [68] to cluster the locations based on a group of geo-tagged Flicker photos. Rather than setting the number of clusters, this algorithm needs to specify a value to determine the density radius, which is difficult to accurately discover proper number for different areas. Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is a widely used algorithm of density-based clustering for geospatial data [69]. In [28], the authors apply DBSCAN clustering algorithm to identify urban AOIs based on Flickr geo-tagged photos. In [3], the authors make use of DBSCAN algorithm to identify clusters based on delivery addresses. Compared to above clustering algorithms, DBSCAN is very robust against outliers, and there are no constraints on the boundaries of clusters. Furthermore, a number of studies have proposed improvements of clustering approaches based on DBSCAN. In [64], the authors apply the algorithm of P-DBSCAN [70] to eliminate noises of geographic coordinates and investigate the tourists' behaviors in Hong Kong. In [71], the authors introduce the algorithm of C-DBSCAN, which defines the constraints based on the background knowledge. In [32], the authors propose M-DBSCAN to reduce the uncertainty of detecting clusters by DBSCAN based on different density and cluster size scales. H-DBSCAN [72] is introduced by improving and integrating DBSCAN and OPTICS [73]. In [74], the authors apply H-DBSCAN algorithm to identify AOIs and interest patterns of tourists from Flickr geo-tagged photos in Vienna. Other clustering methods are also applied to discover clusters of AOIs. In [75], the

authors devise a clustering method for discovering AOIs from image densities and enhanced by the secondary densities of sites adjacent to the images. In [30], the authors propose an adaptive urban clustering method to discover points of interest (POIs) based on different granularities.

Grid-based aggregation. In [76], the authors map coordinates into a grid cell and defined temporal constraints to discover AOIs. In [77], the authors collect geo-tagged photos with location names and conduct clustering by Delaunay triangulation, then POI was recognized as the average coordinates located in the cluster. In [78], the authors propose a grid-based algorithm to solve the problem of discovering geometries of interest (GOIs) of moving objects based on GPS trajectories.

Besides above three groups of approaches, we also have reviewed other single-AOI detection algorithms. In [79], the authors introduce a grid-based Integer Linear Programming (ILP) model to discover AOIs. In [31], the authors propose an algorithm of G-ROI for discovering ROIs on multiple social media datasets. The G-ROI contains two steps of reduction and selection, and achieves higher F_1 score compared with other methods.

The boundary of a cluster is often constructed by a convex hull [80]. Given a set of points on a Cartesian plane, the convex hull of these points is represented by the external polygon on them [65]. In order to match boundaries better and reduce blank parts for polygon construction, the alpha-shape [81] and other approaches of building concave hull are widely used [78]. The algorithm of alpha-shape is built on the base of Delaunay triangulation and can be used to construct polygons with different shapes flexibly. In [82], the authors propose an algorithm of chi-shape to construct a concave hull, and [28] use this method to construct AOIs based on geo-tagged Flicker photos.

For existing studies related to AOI detection, they focused on detecting one partic-

ular AOI. And the research gap is that these studies detect AOIs one at a time and ignore their spatial dependency. This would end up with inconsistent results, e.g., AOIs with overlapping spatial boundaries. To address this issue, a new approach is proposed to detect multiple AOIs simultaneously and solve the multi-AOIs detection problem. In summary, our study is different from earlier research in two ways. First, our approach fully utilizes mutual exclusion of different AOIs and combines multiple single-AOI detection algorithms together; Second, instead of focusing on detecting one particular AOI, we propose an optimization model which can detects multi-AOIs simultaneously.

By following the literature review of multi-AOIs using OFD data, data mining of building footprints within AOIs is described in the next section. A series of deep learning-based approaches using remote sensing images are described in detail.

2.3 POI Detection Using Remote Sensing Images

Over the past few decades with the growth of remote sensing sensors, remote sensing data painted a detailed picture of the urban environment and made the classification and extraction of buildings and other artificial objects possible [83].

Buildings represent an important part of the city. Automatic and accurate extraction of building boundaries can benefit urbanization planning, disaster management, and environmental management, and has been extensively researched over decades [36, 37, 84, 85]. It is seen as the issue of detecting and extracting building areas from images based on techniques of image processing and computer vision [35, 86].

Many studies of building extraction are on the base of traditional image processing methods. In [87], the authors proposed and compared the supervised and unsupervised methods of segmentation in combination with the algorithm of random forest

for extracting buildings from remote sensing data. In [88], the authors proposed the indices of edge patterns and shadow lines as candidates using segmentation models, and different classifiers of machine learning are employed to identify buildings. In [89], the authors developed a new method of fuzzy landscape generation, which is used to detect the building's directional space relation and shade for automatic building extraction.

The term Deep Learning (DL) or Deep Neural Network (DNN) refers to Artificial Neural Networks (ANN) with multi layers. Over the past few years, it has been considered to be one of the most powerful tools, and has become very popular in different research fields due to its capability of dealing with a huge amount of data. Deep Learning has also been broadly used in different remote sensing applications, e.g., object detection [90, 91], scene classification [92], land use mapping [93], etc. Deep learning is a specific branch of Machine Learning (ML), which uses a multiple layer structure to progressively extract higher-level features from raw inputs [94]. And DNN combines operations of linear and non-linear for encoding deep features of inputs [36].

One of the most popular DNNs is the Convolutional Neural Network (CNN). CNN is a class of DNN with multiple connected layers (convolutional layer, non-linearity layer, pooling layer and fully connected layer) for feature extraction with different levels [95]. CNN is able to achieve better performance, particularly in the field of image processing [96]. As deep learning technology develops, many novel technologies of semantic segmentation are developed. CNN's structure with multiple layers is able to learn image features from different levels automatically, which is useful to classify images and promote the growth of semantic segmentation. However, image classification of most CNN models is the patch-based method, and pixels are classified by analyzing the area surrounding each pixel. This method is relatively inefficient since patches are overlapping [97].

In [98], the authors proposed a Fully Convolutional Network (FCN) to address the problem of semantic image segmentation, which can categorize each pixel based on abstract features. In comparison to CNN, FCN is able to deal with images of multiple sizes and improve processing efficiency. Based on the better performance, varieties of methods were developed on the base of FCN. In [97], the authors proposed an FCN-based U-Net structure, which achieves high performance on medical image segmentation. In [99], the authors proposed the structure of DeepLab, which utilizes operations of deconvolution and up-sampling, and expands deep CNN. These two methods focus on the improvement of FCN to obtain a better segmentation.

Then, more models with the structure of encoder-decoder are proposed, which enhance the up-sampling part [100]. Noh *et al.* [101] proposed the structure of DeconvNet, which converts the process of up-sampling to a deconvolution process. CEDN, proposed by Yang *et al.* [102], is developed on the base of encoder-decoder architecture and used for the detection of edges. Badrinarayanan *et al.* [103] proposed the structure of SegNet, which deletes layers with fully connection and uses max-pooling up-sampling. In SegNet, every pooling layer can output two feature maps, namely, the following layer and the decoder of up-sampling. In Marmanis *et al.* [104], the authors applied FCN and SegNet to detect boundaries by semantic image segmentation, and this model achieves high performance of the segmentation. Shi *et al.* [105] developed a novel change detection network, DSAMNet, which integrates the convolutional block attention module for more discriminative features on both spatial-wise and channel wise to achieve fine-grained bitemporal change detection. He *et al.* [106] developed an urban tree specific sub-pixel mapping architecture with deep learning approach to generate 2m fine-scale urban tree cover products from 10m Sentinel-2 images. Deep learning models can achieve better performance than basic machine learning models, therefore, recent studies on semantic image segmentation mostly focus on deep learning networks, and following with some optimization operations to achieve good outcomes [107, 108].

According to recent studies, deep learning-based semantic segmentation models are also used to extract buildings based on remote sensing images [36, 37, 109, 110]. For instance, Maggiori *et al.* [111] used FCN models to achieve the semantic segmentation of buildings from multispectral remote sensing data. In [112], the authors proposed a building extraction model on the base of Res-U-Net structure in combination with guided filters. Sahu *et al.* [110] adopted the standard U-Net architecture with batch normalization layers to extract buildings from remote sensing images. The results are then fed into a post-processing pipeline to separate the connected buildings and convert them into vector formats. In [109], the authors proposed a multi-loss neural network with an attention block based on U-Net, which can extract buildings with high accuracy. And the attention block shows clear advantages in extracting features and detecting edges of images. Chen *et al.* [113] developed a shifted windows transformer as a backbone for building extraction from satellite remote sensing images. These studies show the excellent performance of semantic segmentation models of building extraction.

For extensive studies of building footprint extraction, they recognize the building roofs as the building footprints, which can lead to errors for high-rise buildings due to different sensor view angles of the satellites. Our study can solve this issue which has been ignored in those studies and our method is different from them in various aspects. First, instead of detecting POI boundaries by VGI data, our approach tries to extract building footprints to represent POI boundaries and parse POI names from OFD dataset; Second, most of existing studies extract building roofs as building footprints, while in our study, we introduce an offset vector method to extract building footprints based on the boundaries of the building roofs and the whole building shapes; Third, existing public datasets only label building footprints, while in this research, we labelled both the building roofs and the whole building shapes based on remote sensing images to meet our research objective; Fourth, an end-to-end deep CNN, that is, a multi-task Res-U-Net model with attention

mechanism, is proposed for high-rise building footprint segmentation at pixel level with remote sensing images; Finally, in the proposed model, the attention mechanism shows obvious advantages in extracting features and detecting edges of images, the multi-task strategy helps to extract the building roofs and the whole building shapes together, and the encoder is shared by both the decoders.

By following the literature review for the extraction of building footprints using remote sensing images, data mining of a road network within residential complexes is described in the next section. Different approaches of road extraction are reviewed based on multi-source geospatial data.

2.4 Road Extraction Using Multi-Source Data

Automatic road extraction is an important basis of intelligent transportation system and has been extensively studied in the past few years [114]. According to the input data modality, we divide the existing methods into four main categories and describe the related work of each category.

2.4.1 Remote Sensing Image-based Road Extraction

The rapid evolution of remote sensing technologies enables researchers to easily access huge amounts of high-quality remote sensing data [1, 10]. For existing studies, various approaches were developed to extract road network by using remote sensing data. Previous studies often used traditional machine learning methods to detect road regions based on hand-crafted features including colour and texture [115–117]. While most of them can only be applied into the specific scenarios. Recently, with the better performance of deep learning models in computer vision field, more and

more models have also been utilized for extracting traffic roads from remote sensing images [1]. In [118], the authors developed a new encoder-decoder deep learning model, named CasNet, which is utilized for extracting roads and center lines simultaneously using remote sensing images. Zhang *et al.* [119] developed a road region extraction method by integrating residual learning and U-Net using remote sensing images. This method achieved the best performance among all comparing methods. In [114], the authors developed a novel encoder-decoder deep learning model called D-LinkNet for semantic segmentation. The proposed method uses strategies of dilated convolution and pre-trained encoder to extract road network. And this method achieved the best performance of IoU in the road extraction of CVPR DeepGlobe 2018. In [120], the authors developed a model with the encoder-decoder structure for road extraction, which consists of residual blocks and skip connections for the encoder, and convolutional layers for the decoder. The proposed model performed the best among other state-of-the-art models. In [121], the authors developed a novel network named DRR Net for semantic segmentation using remote sensing images. The proposed model consists of several DRR modules for extracting diverse roads to alleviate the problem of class imbalance. Zhang *et al.* [122] developed an approach to extract road network on the base of Fully Convolutional Networks (FCNs) with an integration strategy to address the uneven distribution of roads and backgrounds of remote sensing images. Even though considerable progress has been achieved in the field of road extraction, complicated scenarios still do not perform well, particularly for the extreme occlusion. It is not easy to deal with road extraction perfectly only based on colour information of remote sensing data. Hence, additional supplementary information would have to be extraction from multiple data sources.

2.4.2 Trajectory-based Road Extraction

In general, a geographic area with large amounts of vehicle trajectory information is more likely to be a road. Therefore, more and more studies tried to extract road network using crowd-sourced trajectories. Due to the GPS noises of trajectory data, previous studies mainly concentrated on how to remove these noises and uncertainties. For existing studies, we divide these approaches into four broad categories. (1) Clustering-based approach. For this approach, a cluster of road ends is generated based on directional similarity and geographic spatial distance, then road ends are connected into road segments from GPS traces [123–125]. In [124], the authors developed a framework that uses the rich knowledge gathered from GPS trajectories to generate up-to-date maps. In comparison with the best performing approaches, this method can generate maps efficiently at the city level. Stanojevic *et al.* [125] proposed two efficient algorithms (online algorithm and offline algorithm) for generating maps based on GPS trajectories. (2) Trace-merging based approach. For this approach, all GPS trajectories are first scanned in sequence, and for every one of them, either merges it with an existing road link or constructs an extra road link otherwise [126, 127]. In [126], the authors developed an approach for converting the original GPS trajectory data of everyday vehicles into the connected road network automatically. Niehoefer *et al.* [127] developed an approach that can generate maps automatically, and the working mechanism is based on GPS tracking contributions from random individuals. (3) Kernel Density Estimation (KDE) based approach. For this approach, KDE is applied to the GPS data and then a map can be generated by image processing techniques [38, 128]. Biagioni *et al.* [38] proposed a novel hybrid map inference approach, which combined several novel innovation points from the existing algorithms. And the proposed method performed the best among all baseline models. Based on the topological idea of the Morse theory, Wang *et al.* [128] developed a new approach to reconstruct maps. Using the topological simplification

and Morse theory enables them to address noise and non-homogeneous sampling problems. (4) Deep learning-based approach. For this approach, Features are first extracted from GPS trajectories, and then rendered as image layers and fed into deep neural networks. In [39], the authors developed a new method for map generation on the base of deep learning methods. In this framework, features are extracted from GPS trajectories into spatial and transition views, and road centerlines are predicted from a convolutional deep neural network called T2RNet. Despite the use of various techniques, the problem of GPS noise remains and the performance of road extraction is not good enough because of the information limitations of crowd sourced trajectories.

2.4.3 LiDAR-based Road Extraction

Light Detection and Ranging (LiDAR) is a kind of remote sensing technology, which can generate accurate digital 3D model by capturing point cloud data. While different from remote sensing images, LiDAR data contains distance information and different objects have different laser reflectivity. Due to advantages of LiDAR, roads are generally recognized as the flatness from the aerial viewpoint, which can be used to differentiate roads from trees and buildings [1]. For existing studies [129–131], various approaches were used to detect road network by using LiDAR data. For instance, Zhang *et al.* [131] developed a method to detect road and road-edge with fast processing speed and reliable performance. The proposed method has been verified by a public dataset to show its robustness and efficiency. In [132], the authors developed a road centerline detection approach on the base of the multi-feature strategy. For this method, the main idea is to detect possible road centerlines and delete joined non-road components from roads. In [130], the authors proposed an approach for extracting road centerlines by LiDAR data. The proposed method contains three core algorithms, and was proved to be efficient for the road centerline extraction.

Even though there has been considerable progress, LiDAR-based road extraction still exists some challenging problems such as the intensity of reflection and data noise. Because of the sparsity of LiDAR data and noisy points, the performance of existing methods is relatively poor in complex scenarios [133].

2.4.4 Multi-Sourced Data Road Extraction

As described above, every data source has its own advantages and disadvantages, therefore, it is reasonable to combine multi-sourced data for efficient road extraction. For existing studies, the researchers have proposed numerous approaches for road extraction using both LiDAR and remote sensing images [1]. For instance, Hu *et al.* [134] proposed an approach of semantic segmentation and image analysis to detect the roads and background targets based on fusing both remote sensing images and LiDAR data. And multiple strategies for fusing LiDAR data and images have been applied to this method. However, both LiDAR data and remote sensing images cannot provide enough information for road extraction with heavy occlusion of trees, therefore, more and more studies incorporated remote sensing data and crowd sourced trajectories to extract road network. In [40], the authors developed an approach for improving the road extraction using both remote sensing data and GPS trajectories. A novel strategy of 1D transpose convolution has been applied to the proposed approach, and its effectiveness is verified to improve the model performance. In [41], the authors developed a novel transfer learning approach for constructing road map intersections and links using GPS trajectories and remote sensing data. In [135], the authors developed a new network of DeepDualMapper on the base of CNN, which integrates remote sensing data and GPS traces for road extraction. In this study, the authors designed a gated fusion module for further improving the accuracy of road extraction. In [136], the authors developed a multi-modal road extraction approach to boost the performance of pixel-level ex-

traction using crowd-sourced GPS trajectories and aerial images. In [1], the authors developed a network of CMMPNet to take advantage of the complementarities of remote sensing images and GPS traces for road extraction. Even though progress has been made through fusion approaches, this does not take fully advantage of the complementarities of multiple sourced data and require more efficient models to extract road network [1]. For existing studies, they either connected different features directly or averaged the predictions of multiple models for data fusion, therefore, multiple sourced data were not fully utilized [135].

For extensive studies of road extraction, they focus on traffic roads and most of the GPS trajectories are collected by taxi. While for interior roads, most of them are within gated residential complexes in Chinese cities, and taxis are not allowed to enter in some gated residential complexes. For OFD services, deliverers can enter the gated residential complexes and deliver food to customers' doorsteps. Therefore, GPS trajectories generated by riders can be used to supplement data sources for interior road extraction.

Chapter 3

Geolocation Inference Using Twitter Data

[1]This chapter presents a method of geolocation inference based on the potential location-related metadata of tweets. A crude form of geographic coordinate information can be obtained from every tweet's bounding box, while location-related information can be extracted from the textual content, user location, and place labels via NER techniques. Three coordinate datasets of the United States counties are built and used as the coordinate references. Models with different data sources have been employed to predict the geolocations of the tweets related to COVID-19 in the contiguous United States. Results show that the models with four data sources perform better than other models.

[1]Parts of this chapter have been published in Li, B., Chen, Z., and Lim, S. (2021). Geolocation Inference Using Twitter Data: A Case Study of COVID-19 in the Contiguous United States. *Communications in Computer and Information Science*, vol 1411. Springer, Cham. https://doi.org/10.1007/978-3-030-76374-9_8.

3.1 Introduction

In December 2019, the initial cases of pneumonia associated with a novel coronavirus occurred in Wuhan City, China [137]. However, measures to control the spread of the virus were not implemented effectively to keep its spread within China [138]. Since then, the coronavirus disease 2019 (COVID-19) has been rapidly spreading around the world, causing tens of millions of cases around the world [137]. As of July 15th, 2022, almost 566 million (565,606,477) cases have been recorded, including 6,382,616 deaths where 14.27% (91,060,225) of those cases occurred within the United States, including 1,048,232 deaths according to the worldometer coronavirus pandemic tracker [139]. Therefore, an overarching objective of this chapter is to contribute to the identification of spatio-temporal patterns of the COVID-19 pandemic with a particular interest in the United States.

Over the past decade, the Internet has helped revolutionize every aspect of people's lives, and it is not only a source to get information, but also a platform to disseminate personal information [12, 13]. In addition, the development of mobile devices made it easier to send digital information (e.g., texts, location labels, and pictures). At the same time, social media platforms have experienced a tremendous and profound reform. Twitter and Facebook mainly provide basic services, but other types of social media are being used to connect online for different reasons, such as location-based services (e.g., Foursquare and Whrrl), media sharing services (e.g., Instagram, Snapchat, and Flickr), as well as other types of services (e.g., Quora, Medium, and LinkedIn). Users can establish online friendships based on mutual interests and share their everyday life with each other.

Supported by previous studies [14–16], Twitter outshines other platforms regarding social network analysis and event detection because of not only its excellent design, but also its vast user base of different age groups. According to the most up-

to-date Twitter statistics for 2022, its daily active users are around 211 million, which accounts for 23% of the Internet population, and about 500 million tweets are posted every single day [43]. Compared with Instagram and Snapchat regarding the demographics, Twitter is widely used by people of different ages and nearly 63% of them age between 35 and 65 [140]. The large quantity of user-generated contents is employed for data mining in various research areas [12]. Tweets with accurate geographic information can provide significant benefits to event response and monitoring, hence those without geographic information become useless unless geolocation inference is applicable. Accurate prediction of tweets' geolocation can effectively benefit the response and rescue in emergency events [17].

The development of GPS-enabled mobile devices enables users to share and track their locations with accurate geographical coordinates. However, due to the operational complexity and privacy concerns, most users do not turn this function on [18]. As Laylavi *et al.* [19] illustrated, the percentage of tweets with geo-tags accounts for only 2%, which severely limits the development of associated applications. Therefore, accurate geolocation inference of tweets has become an urgent problem in this research field.

Nowadays, disease-related information is increasingly shared in real time through Twitter, while timely data with spatial and temporal information play a significant role in surveillance of an epidemic disease [20, 21]. Every single tweet has its own metadata, which includes its creation time, but under most circumstances, does not contain its created geographical coordinates, hence geolocation inference of tweets is still a critical issue. Real-time data without any geographic information can be almost meaningless for emergency response and surveillance of an epidemic disease. Thus, this chapter aims to develop novel methods to predict the geolocation of non-geotagged tweets based on their own metadata.

In this chapter, models based on multiple attributes of the tweet's metadata are

built to predict the non-geotagged tweets' geolocation. Attributes of textual content, user location, place labels, and bounding box are fully used during the modelling process. The dataset used in this chapter was collected between the 10th and 30th of June 2020. During this time, the US was suffering a severe effect of the COVID-19 pandemic. The development of technologies, including NLP and NER make it easier to extract location entities from textual data.

The main contributions of this chapter are summed up as follows.

- Exploring potential location-related attributes of the tweet's metadata and extracting location entities via NER techniques.
- Three geographic coordinate datasets of counties are used to predict geolocation and the proposed models are built according to different priorities of location-related attributes.
- When the area threshold of the bounding box is set to 10,000 km^2, the best model can successfully predict the geolocation of 90.8% of COVID-19 related tweets with the mean error distance of 4.824 km and the median error distance of 3.233 km.
- The proposed method enhances the granularity of geographic information of tweets and makes the surveillance of COVID-19 effective and efficient.

The rest of this chapter is organized as follows. Section 3.2 presents a brief introduction of Twitter data's structure. Detailed explanation of the proposed method is described in Section 3.3. A case study of the COVID-19 in the contiguous US based on the proposed method is illustrated in Section 3.4. The summary of this chapter is provided in Section 3.5.

3.2 Structure of Twitter Data

Twitter was released in March 2006 and now has about 330 million active users per month. Tweets can be posted by users via this platform. In its early days, every tweet can contain up to 140 characters, but the length of it was doubled in 2017 [141]. This increase provided users more space to express their ideas and saved more time of text compression than before. Every tweet's metadata contains a wealth of information about itself, while it is only visible to developers, not common users. Twitter data can be collected based on Twitter application programming interfaces (APIs) and stored with the format of JavaScript Object Notation (JSON). JSON format is lightweight and easy for both human beings and machines to understand and use. A JSON object contains a key/value pair and is normally enclosed in a pair of curly braces [142]. The structure of Twitter data consists of several objects, including tweet object, user object, coordinates object, place object, and bounding box object, which are all encoded in JSON format. For every tweet, the metadata can tell us its username, textual content, unique identification (ID), created time, and occasionally geographic details of where it was posted. In general, every tweet's metadata contains more than 150 attributes, while only spatio-temporal information related attributes are taken into consideration in our research.

Figure 3.1 shows the spatio-temporal information related attributes in a tweet's metadata. The attribute of "*location*" is an element of the user object and is defined by user himself/herself, therefore, it can be a location that does not exist in the real world or cannot be recognized by computers. Another one is "*geo_enabled*", which means if the current user can attach geographic data or not. This attribute is very important for location-related studies, although it does not contain any essential geographic information.

Both attributes of "*coordinates*" and "*geo*" represent the specific longitude and

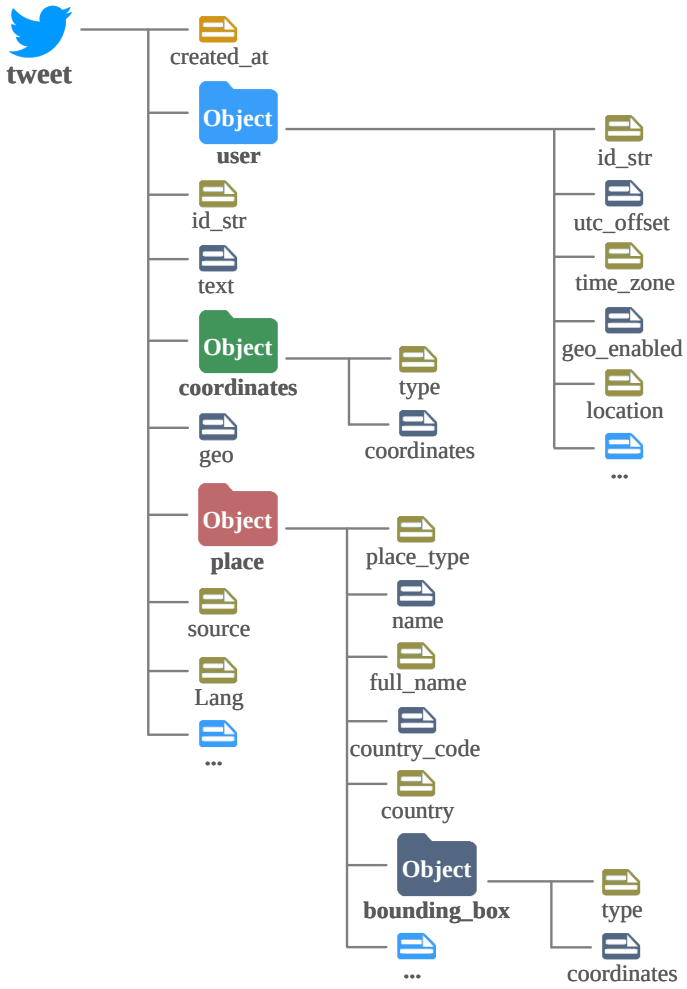


Figure 3.1: Spatio-temporal attributes of a tweet's metadata.

latitude of the tweet's location, as a collection in the form $[longitude, latitude]$. However, "*geo*" has been deprecated according to the twitter official document, hence we used the attribute of "*coordinates*" to acquire accurate geographic coordinates of tweets.

Place object contains various location-related attributes. The attribute "*place_type*" represents the type of location of this place and it has five values to choose from. Table 3.1 shows five values of attribute of "*place_type*" and statistics of our research dataset. For POI, it represents the specific location of a place, e.g., Washington Square Park, while the other four values stand for a certain area. Due to the large

Table 3.1: Typical values and statistics of "place_type" attribute.

Category	Amount	Percentage	Example
POI	119,655	0.96%	Washington Square Park
neighborhood	25,183	0.20%	Downtown Jacksonville, FL
city	10,301,683	82.98%	Los Angeles, CA
admin	1,942,596	15.65%	California, USA
country	26,105	0.21%	Canada

regional extent of city, admin, and country, we used data from only POI and neighborhood. Attributes of "*name*" and "$full\_name$" are two ways to describe the place's names. While "*country_code*" and "*country*" provide the short code and exact name of the country of the place. The attribute of "*bounding_box*" is four lon/lat pairs of each corner of a box that contains the place.

3.3 Methodology

Figure 3.2 plots the workflow to illustrate the architecture of the proposed method of this research. This method is generally divided into three modules. In the first module, real time tweets within a bounding box are collected. Tweets data are initially stored into text files and then read based on JSON format. Then the data enters the preprocessing and geotagging stage, after which a dataset with geo-tagged tweets is created. In the second module, location entities are extracted from textual content, user location and place labels via NER techniques. Combining geometric properties of the place's bounding box, as well as coordinate datasets of gazetteers and digital boundaries of the US, all these data are fed into 16 models to predict tweets' geolocation. Finally, predicted results are evaluated by mean error distance (MED) and median error distance (MDED).

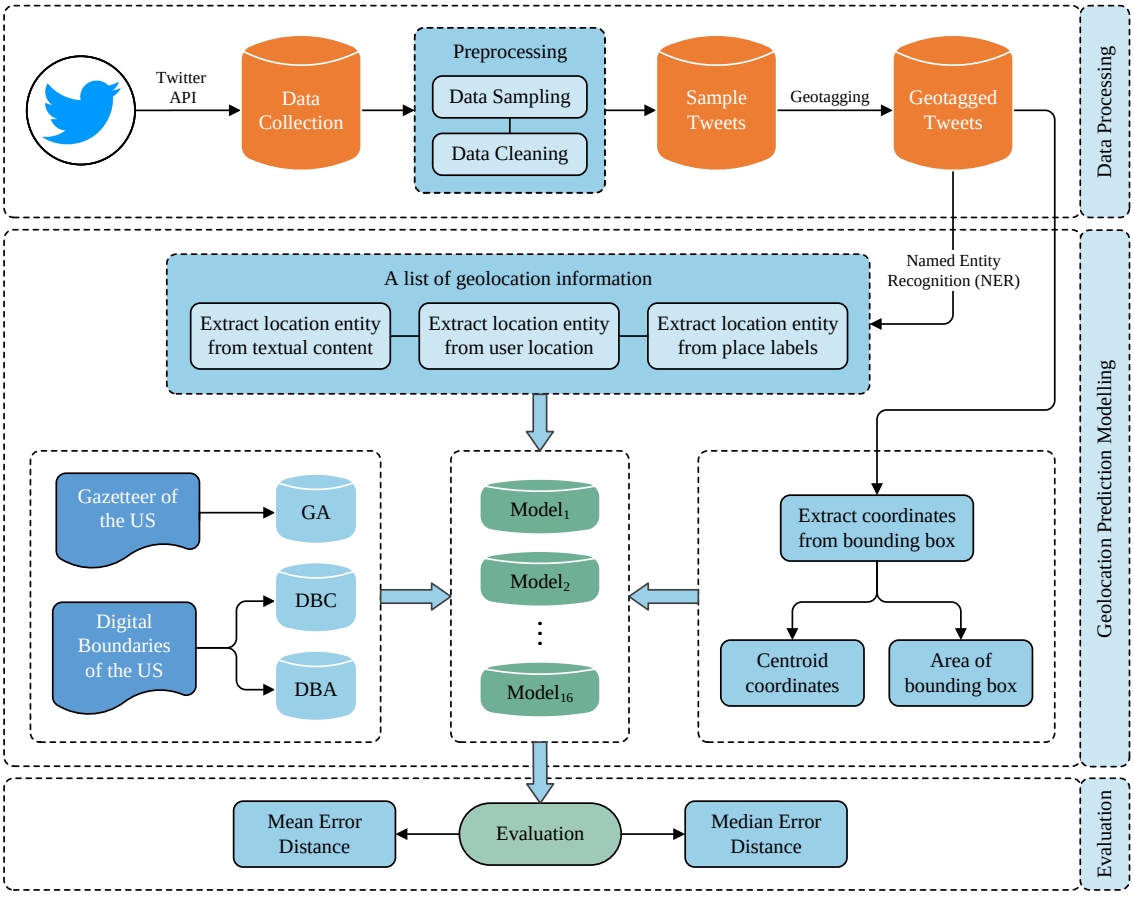


Figure 3.2: Workflow of geolocation inference of tweets [143].

3.3.1 Data Collection

Twitter data can be gathered from both business companies and Twitter API which is available free of charge. As for commercial purchases, the companies can provide both historical and real time tweets from all over the world, but the price is very high. Twitter API can help collect tweets freely, but only real time tweets within the specific bounding box can be collected. Therefore, it normally takes several months to collect the whole research data using Twitter API. In this chapter, data collection was done via Twitter API, and it was implemented by the *tweepy* library of python [19, 144]. The data were collected from June 10th to June 30th, 2020 in the contiguous US during the COVID-19 pandemic. During this period, 12,408,538

unduplicated tweets were collected and stored into local text files. Only tweets located in the area of longitudes from 66°W to 125°W and latitudes from 24°N to 49°N are collected, as shown in Figure 3.3. While within the bounding box, some tweets from Canada, Mexico, and the Bahamas were also included, but excluded in this research.

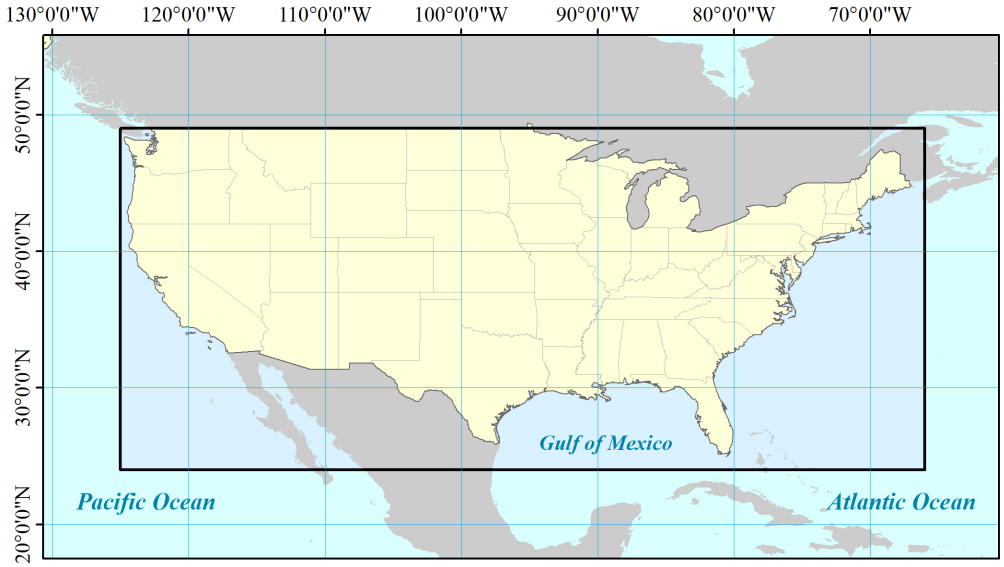


Figure 3.3: Area of data collection.

3.3.2 Data Preprocessing

3.3.2.1 Data Cleaning

In the textual content of every tweet, it often contains noises, including hashtags, mentions, emojis and Uniform Resource Locator (URL) links, hence preprocessing operation is necessary. In this step, we used regular expressions to process textual data. A regular expression is a pattern that attempts to match with input text and can be implemented by python *re* library [145]. URL links started with "*https*://" and "*http*://" were removed from the textual content since they do not contain any

location related information. We replaced unnecessary punctuation marks into a space, and consecutive spaces into one. Marks of user mentions, hashtags, non-English letters as well as stop words were all deleted [144]. As for the user location, it can be modified by users at will, thus the information was processed in the same manner.

3.3.2.2 Data Sampling

A workflow was plotted to illustrate how useless tweets are filtered out and generated a new dataset. The dataset was mainly processed via the python pandas library. Firstly, the method of "*drop_duplicate*" is employed to delete duplicated tweets from the dataset. The attribute of "*lang*" indicates the language used by every tweet, and only English tweets are kept in our study. As noted above, tweets posted outside the contiguous US are also removed from the dataset.

Another problem is that many tweets are meaningless to this chapter, such as those posted by advertisers or spambots. This kind of tweets is mainly posted by computers, therefore, only tweets posted by mobile devices (e.g., iPhone, Android, iPad, and Instagram) are kept, and the attribute of "*source*" was used to implement this function [19, 144]. Then tweets without geo-tags were filtered out and implemented by the "*coordinates*" attribute. Finally, the COVID-19 related tweets were extracted by using the keywords to match the "*text*" attribute of every tweet. We introduced Term Frequency-Inverse Document Frequency (TF-IDF) to get keywords from news articles about the COVID-19 pandemic in the US, and TF-IDF score helped us extract keywords from the related articles [146]. Supported by recent studies [137, 147, 148] and TF-IDF techniques, we used the following keywords: "corona", "coronavirus", "covid", "covid-19", "ncov", "sarscov2", "ncov2019" and "2019ncov" to extract COVID-19 related tweets. Through data sampling, 3,600 corresponding tweets were retrieved from the Twitter dataset. Figure 3.4 shows the

whole data sampling process.

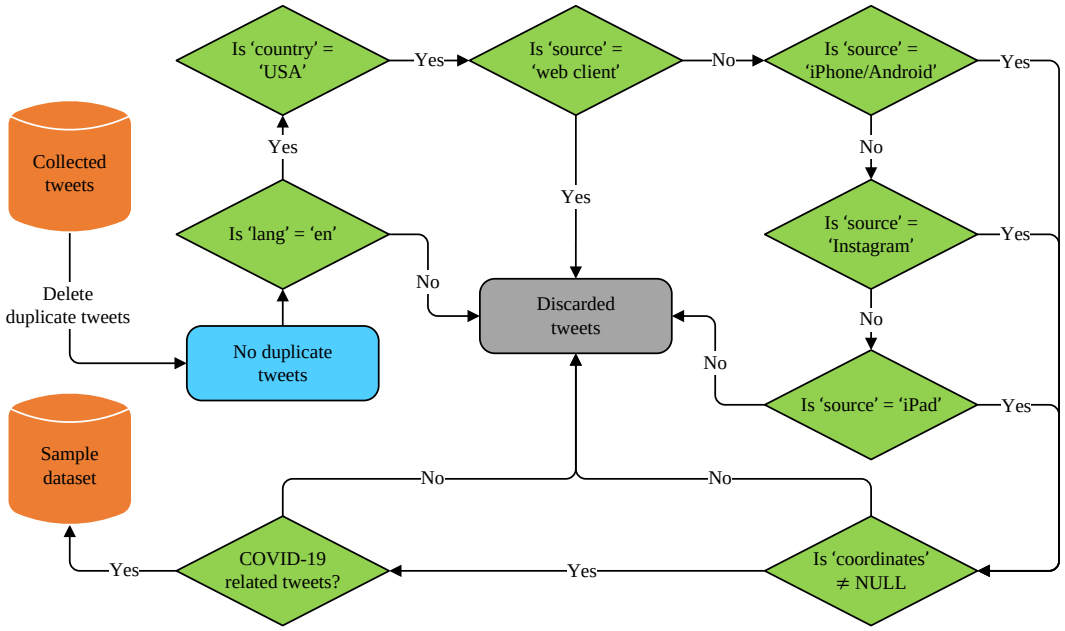


Figure 3.4: Flowchart of data sampling [143].

3.3.3 Location Information Extraction

3.3.3.1 Named Entity Recognition

NER can be used to recognize and classify different types of entities (e.g., location names, person names, and organizations) from unstructured texts. It has been extensively studied in the last few years in machine learning and NLP. While it does not work well on informal texts like tweets since it is usually built on the basis of formal texts [56]. As for this technique, it can help to answer many real-world questions, such as: "*Does a tweet contain the name of a person or does the tweet provide a person's current location?*" In this chapter, we use NER to identify locations from textual content, user location, and place labels of tweets on spaCy.

3.3.3.2 Bounding Box

The bounding box is a specified 4-sided geographic area and matching the tweet's location falling into the area. Unlike other location related geographical metadata, the bounding box contains the accurate lon-lat coordinates of the four points enclosing the place. Due to different types of places, bounding box has different areas. For instance, four points of a bounding box are $Point_1 = (\lambda_1,\ \varphi_1)$, $Point_2 = (\lambda_2,\ \varphi_1)$, $Point_3 = (\lambda_2,\ \varphi_2)$ and $Point_4 = (\lambda_1,\ \varphi_2)$, then Equation (3.1) can be used to calculate the area of this bounding box.

$$S = R^2 \cdot |(\lambda_2 - \lambda_1) \cdot (sin\varphi_2 - sin\varphi_1)|, \tag{3.1}$$

where R refers to the earth radius. λ_1 and λ_2 represent the longitudes of the bounding box, and φ_1 and φ_2 refer to the latitudes of the bounding box. Equation (3.1) can be used to calculate the size of the bounding box. The bounding box's centroid can be reckoned as the predicted location of a tweet, therefore, if the bounding box's area is smaller, it can provide a relatively more accurate prediction. For city, admin and country, the bounding box is too large to be used to predict the geolocation.

3.3.4 Modelling

The location-related information is obtained from the four sources: textual content, location of user profile, place labels, and bounding box. Three coordinate datasets of counties are constructed based on gazetteers and digital boundaries of the US.

Table 3.2: Data fields of US gazetteers.

Field	Description
USPS	United States Postal Service state abbreviation.
GEOID	Unique geographic identifier for each feature.
NAME	Name of the feature.
INTPTLAT	Latitude of the feature in decimal degrees.
INTPTLON	Longitude of the feature in decimal degrees.

3.3.4.1 United States Gazetteers

The national gazetteers of the US were used as the data source and called GA in this chapter. It is a dataset including county's names and information related to geography in the US. This data is provided by the United States Census Bureau, and researchers can download it for free [149]. There are totally ten fields in the dataset, and some of them are displayed in Table 3.2. The field of "NAME" can provide duplicate names, but they locate in different states which means they have different values of "USPS". Fields of "INTPTLAT" and "INTPTLON", respectively, refer to latitude and longitude of the specific county.

3.3.4.2 Digital Boundaries of the United States

Digital boundaries of the US are in the format of Environmental Systems Research Institute (ESRI) *lpk*. This group layer can be freely downloaded from the website of ESRI and presents counties of the US in the 50 states, the District of Columbia, and Puerto Rico. The detailed datasets are represented as polygons with over 40 fields [150].

In this chapter, we only used digital boundaries of US counties due to the coarse granularity of location inference based on the city and state level. In order to obtain

geographic coordinates of each county, we developed two ways to compute them and named them Digital Boundary's Centroid (DBC) and Digital Boundary's Average (DBA). DBC is calculated based on geometric properties of every county's polygon, and the value can be calculated by the centroid of the polygon. On the other hand, DBA is calculated by tweets falling into the county's polygon and the value can be calculated by their average latitude and longitude. For instance, suppose there are m counties in the contiguous US which are $County_1, \cdots, County_j, \cdots, County_m$ and $P\_tweet_1 = (\lambda_1,\ \varphi_1), \cdots, P\_tweet_i = (\lambda_i,\ \varphi_i), \cdots, P\_tweet_n = (\lambda_n,\ \varphi_n)$ are geographic coordinates of n tweets located in $County_j$, then the predicted coordinates of $County_j(P\_county_j)$ can be calculated by Equation (3.2). This method can help compute the average longitude and latitude of geotagged tweets falling into the county's polygon.

$$P\_county_j = \left(\bar{\lambda},\ \bar{\varphi}\right) = \left(\frac{\sum_{i=1}^{n} \lambda_i}{n},\ \frac{\sum_{i=1}^{n} \varphi_i}{n}\right). \tag{3.2}$$

After calculating all polygons' coordinates based on DBA and DBC, Figure 3.5 shows the distribution of distances between DBA and DBC of counties in the contiguous US. This figure illustrates that the distance difference is less than 20 km in most countries, especially for the smaller ones, while for some larger counties in the west and northeast corner, the difference is about 40 km or more. Smaller distance difference means two predicted methods are close to each other. When the distance difference is larger, the better method of coordinates prediction can achieve a better performance.

3.3.4.3 Modelling

As demonstrated in Figure 3.2, the model is on the basis of four location-related attributes of the tweet's metadata: textual content (T), user location (U), place

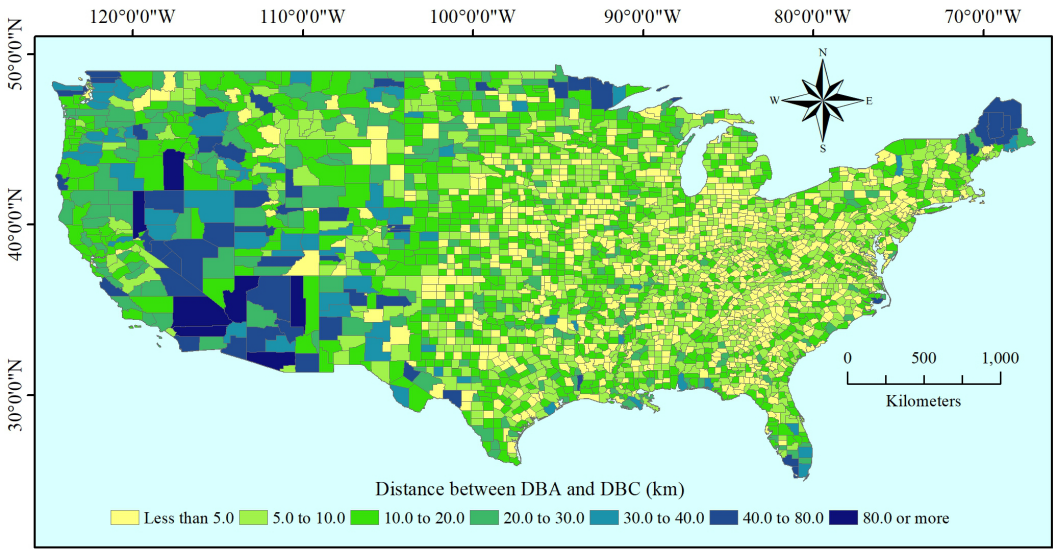


Figure 3.5: Distribution of distance difference of counties in the contiguous US.

label (P) and bounding box (B). Location entities are extracted from T, U, and P by NER techniques, and then query them through coordinate datasets of GA, DBC, and DBA. Equation (3.3) illustrates how the three predicted matrices are computed.

$$\begin{bmatrix} Text_1 & UserLoc_1 & Place_1 \\ \vdots & \vdots & \vdots \\ Text_i & UserLoc_i & Place_i \\ \vdots & \vdots & \vdots \\ Text_n & UserLoc_n & Place_n \end{bmatrix} \xrightarrow[GA,\,DBC,\,DBA]{NER} \left\{ \begin{bmatrix} T_{GA_1} & U_{GA_1} & P_{GA_1} \\ \vdots & \vdots & \vdots \\ T_{GA_i} & U_{GA_i} & P_{GA_i} \\ \vdots & \vdots & \vdots \\ T_{GA_n} & U_{GA_n} & P_{GA_n} \end{bmatrix} \begin{bmatrix} T_{DBC_1} & U_{DBC_1} & P_{DBC_1} \\ \vdots & \vdots & \vdots \\ T_{DBC_i} & U_{DBC_i} & P_{DBC_i} \\ \vdots & \vdots & \vdots \\ T_{DBC_n} & U_{DBC_n} & P_{DBC_n} \end{bmatrix} \begin{bmatrix} T_{DBA_1} & U_{DBA_1} & P_{DBA_1} \\ \vdots & \vdots & \vdots \\ T_{DBA_i} & U_{DBA_i} & P_{DBA_i} \\ \vdots & \vdots & \vdots \\ T_{DBA_n} & U_{DBA_n} & P_{DBA_n} \end{bmatrix} \right. , \quad (3.3)$$

where $Text_i$, $UserLoc_i$, and $Place_i$ are respectively textual content, user location, and place label of a tweet. T_{GA_i}, U_{GA_i}, and P_{GA_i} are predicted coordinates corresponding to $Text_i$, $UserLoc_i$, and $Place_i$, respectively, based on GA. T_{DBC_i}, U_{DBC_i}, and P_{DBC_i} are predicted coordinates corresponding to $Text_i$, $UserLoc_i$, and $Place_i$, respectively, based on DBC. T_{DBA_i}, U_{DBA_i}, and P_{DBA_i} are predicted coordinates corresponding to $Text_i$, $UserLoc_i$, and $Place_i$, respectively, based on DBA.

In Equation (3.3), the value will be stored as "null" if there is no county found based on NER. When we use NER to query the specific county's name, sometimes several results will be found since there are duplicate names of different counties. Therefore, the distance between the predicted point and centroid of the tweet's bounding box should be computed first, if it is within the specific threshold range, the predicted

point can be reckoned as a valid result, otherwise will be discarded.

T_{GA_i}, U_{GA_i}, and P_{GA_i} can be "null" if corresponding counties are not found in GA. T_{DBC_i}, U_{DBC_i}, and U_{DBC_i} can be "null" if corresponding counties are not found in DBC. T_{DBA_i}, U_{DBA_i}, and U_{DBA_i} can be "null" if corresponding counties are not found in DBC. Equation (3.4) shows how the area and centroid's coordinates are computed by the tweet's bounding box.

$$\begin{bmatrix} BBox_1 \\ \vdots \\ BBox_i \\ \vdots \\ BBox_n \end{bmatrix} \xrightarrow{area, centroid} \begin{bmatrix} B_{AREA_1} & B_{CEN_1} \\ \vdots & \vdots \\ B_{AREA_i} & B_{CEN_i} \\ \vdots & \vdots \\ B_{AREA_n} & B_{CEN_n} \end{bmatrix}, \tag{3.4}$$

where $BBox_i$ is the tweet's bounding box. B_{AREA_i} and B_{CEN_i} are the area and centroid's lon-lat coordinates of $BBox_i$, respectively.

Because every tweet has the attribute of bounding box, every model in our study contains this attribute and is placed in the last position. UPTB is one model and designed according to the order of U, P, T, and B. Figure 3.6 illustrates a flow diagram of how UPTB works based on GA.

As shown in this flow chart, n elements are traversed in the outermost. Then, if T_{GA_i} is not "null", it is passed directly to the UPTB dataset, otherwise indicated by U_{GA_i}. If U_{GA_i} is not "null", it is passed directly to the UPTB dataset, otherwise indicated by P_{GA_i}. If P_{GA_i} is not "null", it is passed directly to the UPTB dataset, otherwise indicated by B_{AREA_i}. If the value of B_{AREA_i} is not more than the $Area\_Threshold$, B_{CEN_i} is passed to the UPTB dataset and then a new loop starts, otherwise a new loop starts directly and the final result will be set as "null". When the predicted result is "null", it means geo coordinates of this tweet cannot be predicted based on

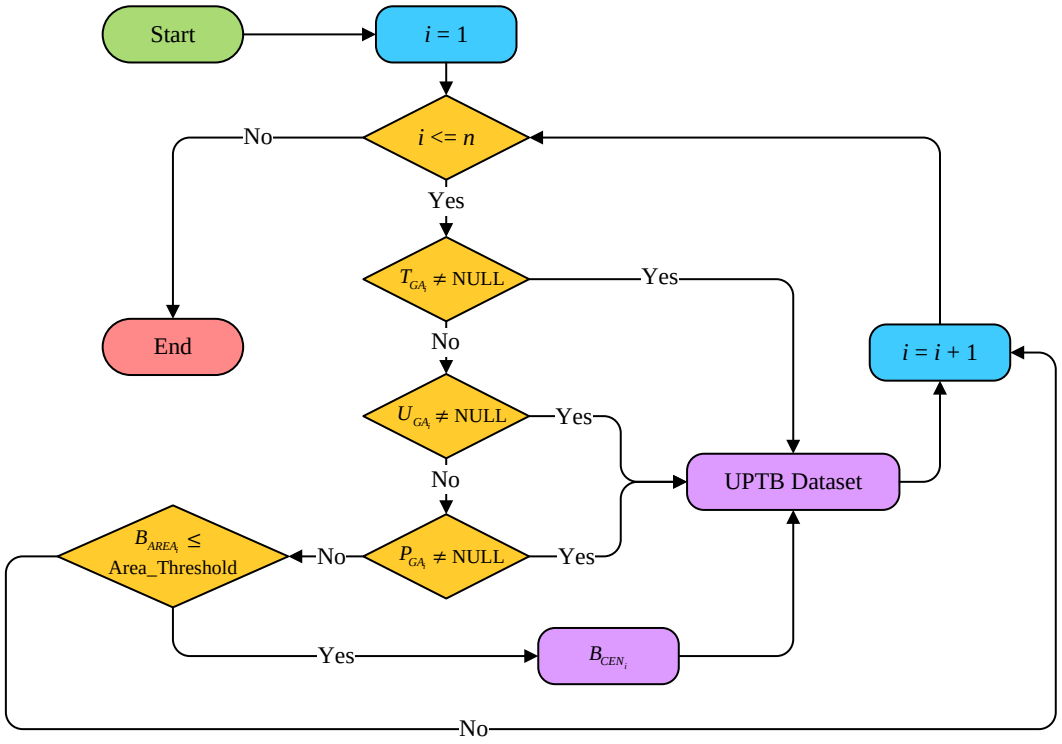


Figure 3.6: Working principle of UPTB based on GA.

this model.

The other models are implemented with the same mechanism. That is, six models (i.e., TUPB, TPUB, UTPB, UPTB, PUTB, and PTUB) contain four parameters, six models (i.e., TUB, TPB, UTB, UPB, PTB, and PUB) contains three parameters, three models (i.e., TB, UB, and PB) contain two parameters and one model (i.e., B) contains merely one parameter. A total of 16 models are implemented in this chapter.

3.4 Experiments

We applied models mentioned in Section 3.3 to the sample dataset and evaluated their performance based on different metrics.

Table 3.3: Statistical information about Twitter dataset.

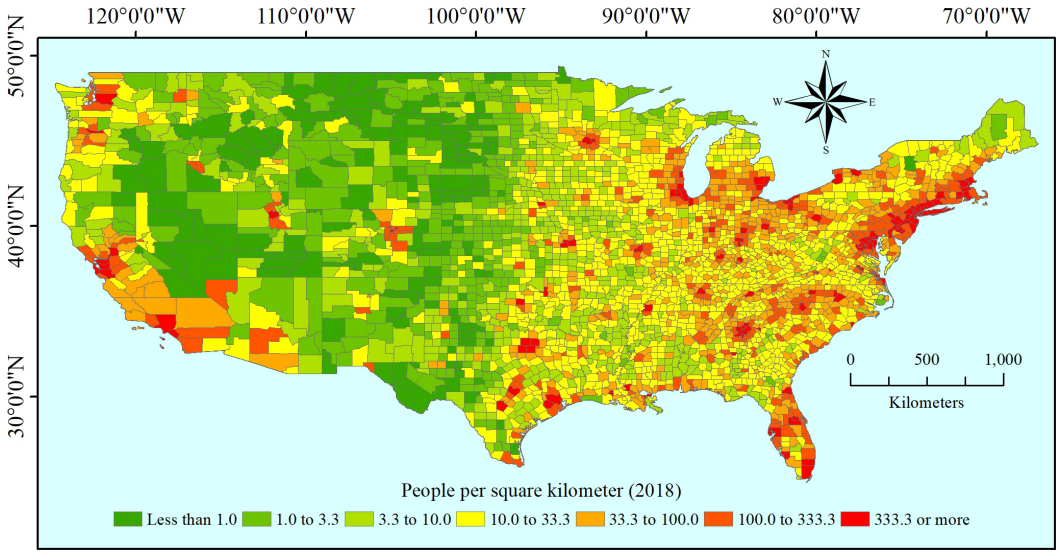
Item	Content
Database size	61.0 GB
Date of data gathering	2020.06.10−2020.06.30
Total number of tweets	12,415,222 tweets
Total number of unique tweets	12,408,538 tweets
Total number of tweets from mobile devices	11,475,982 tweets
Total number of tweets from Instagram	401,610 (3.24%)
Total number of English tweets	10,056,767 tweets
Number of geo-tagged tweets	758,946 tweets (6.11%)
Number of geo-tagged tweets related to COVID-19	3,600 tweets (0.03%)

3.4.1 Research Data

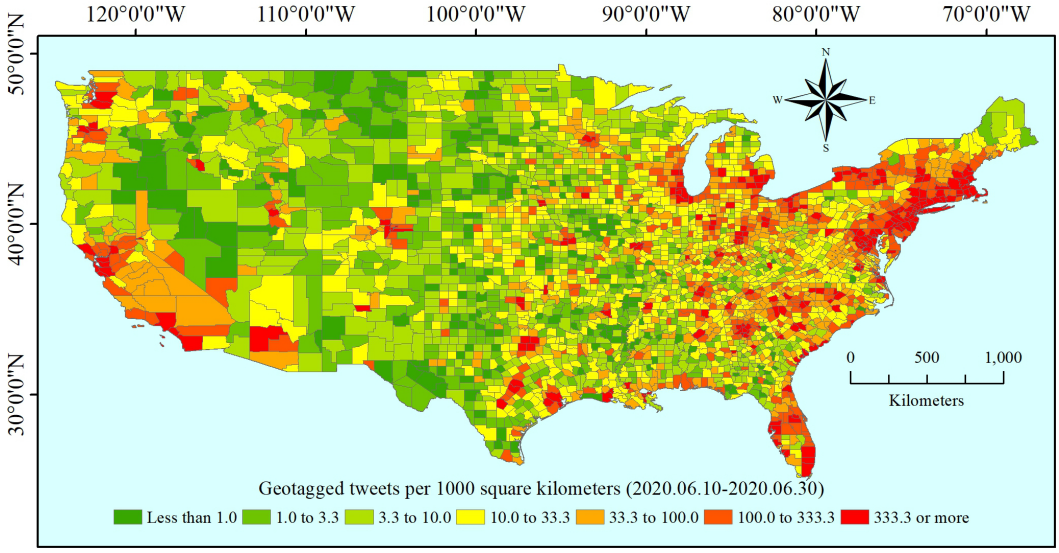
Table 3.3 shows the Twitter dataset that we used in this chapter. We collected these tweets from 10th to 30th of June 2020 in the contiguous US during the COVID-19 pandemic spreading around the world. The total number of collected tweets are around 12.42 million and tweets with geo-tags account for 6.11%. Only geo-tagged tweets related to COVID-19 are applied to the models described in Section 3.3, and the number is 3,600.

As shown in Table 3.3, geo-tagged tweets account for 6.11% of the total Twitter dataset. These tweets were extracted, then plotted with digital boundaries of the contiguous US. Figure 3.7(a) [149] shows the population distribution of the contiguous US counties (i.e., people per square kilometer of 2018), and Figure 3.7(b) shows the geo-tagged tweets distribution based on the contiguous US counties (i.e., geotagged tweets per 1,000 square kilometers between June 10th and June 30th, 2020).

In statistics, the Pearson's Correlation Coefficient (PCC) is a statistic that measures



(a) Population distribution in the contiguous US.



(b) Tweets distribution in the contiguous US.

Figure 3.7: Population and tweets distribution in the contiguous US.

linear correlation between two variables. The value range of PCC is between -1 and 1, and the higher the value, the better the positive linear correction. Equation (3.5) shows how to calculate PCC from two paired data $(x_1,\ y_1), \cdots, (x_i,\ y_i), \cdots, (x_n,\ y_n)$ consisting of n pairs.

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}, \tag{3.5}$$

In this chapter, x_i means people per square kilometer in every county, and y_i means tweets per 1,000 square kilometers in every county. PCC of the two variables in this chapter is 0.88, which indicates a strong positive correlation. Figure 3.7 also shows that population distribution and tweets distribution have a high correlation, hence we can detect real world events based on geo-tagged tweets or tweets with predicted geolocation.

3.4.2 Evaluation Metrics

Models' performance can be evaluated by the distance between the predicated geolocation and the real geolocation of a tweet. The actual distance between two points on the earth's surface can be calculated by the great circle distance. For instance, the great circle distance of two points, $p_1 = (\lambda_1,\ \varphi_1)$ and $p_2 = (\lambda_2,\ \varphi_2)$, can be calculated by Equation (3.6).

$$\begin{aligned} &Dist(p_1,\ p_2) \\ &= 2 \cdot R \cdot arcsin\left(\sqrt{sin^2\left(\frac{\varphi_2 - \varphi_1}{2}\right) + cos(\varphi_1) \cdot cos(\varphi_2) \cdot sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right), \end{aligned} \tag{3.6}$$

where R is the earth radius. λ_1 and λ_2 refer to the longitudes of points, and φ_1 and φ_2 refer to the latitudes of points.

Mean error distance (MED) and median error distance (MDED) are two metrics to evaluate models in our research, and are implemented by Equations (3.7) and (3.8), respectively.

$$MED = \frac{1}{n_{tweets}} \sum_{i=1}^{n_{tweets}} Dist\left(\hat{p}_i,\ p_i\right), \tag{3.7}$$

$$MDED = median_{i=1}^{n_{tweets}} Dist\left(\hat{p}_i,\ p_i\right), \tag{3.8}$$

where $\hat{p}_i$ represents the predicted geolocation and p_i refers to the real geolocation of a tweet.

The tweet's metadata indicates that the value of bounding box is always not null, therefore, it can be used to predict the geo coordinates of the tweet. But its area varies a lot among different tweets and the error distance can be affected dramatically. Figure 3.8 shows the variation of MED and its percentage based on different area thresholds of the bounding box. For example, if the area threshold is set to 1,000,000 km^2, almost 100% of tweets can predict the geo coordinates, but the MED is almost 25 km. When the area threshold is set to 5,000 km^2, almost 90% of tweets can be valid to predict, and the MED improves to 5 km. As shown in Figure 3.8, when the area threshold is set to 5,000 km^2 and 10,000 km^2, the MED and percentage can achieve a relatively better performance, thus the following experiments were conducted by these two values.

Sometimes users mention some other location names rather than the place where tweets are posted. But in most cases, users are more likely to be within or around the place. In addition to this, there often exist duplicate names of different counties in the datasets of GA, DBC, and DBA. Therefore, sometimes several counties were extracted by NER from a tweet. To resolve this issue, we only focus on the predicted location in the bounding box and the distance between it and the bounding box's centroid is within the specific range. In this chapter, we chose the distance threshold from 1 km to 10 km. For example, when the distance threshold is set to 6 km, only the first result with distance of predicted point and bounding box's centroid no

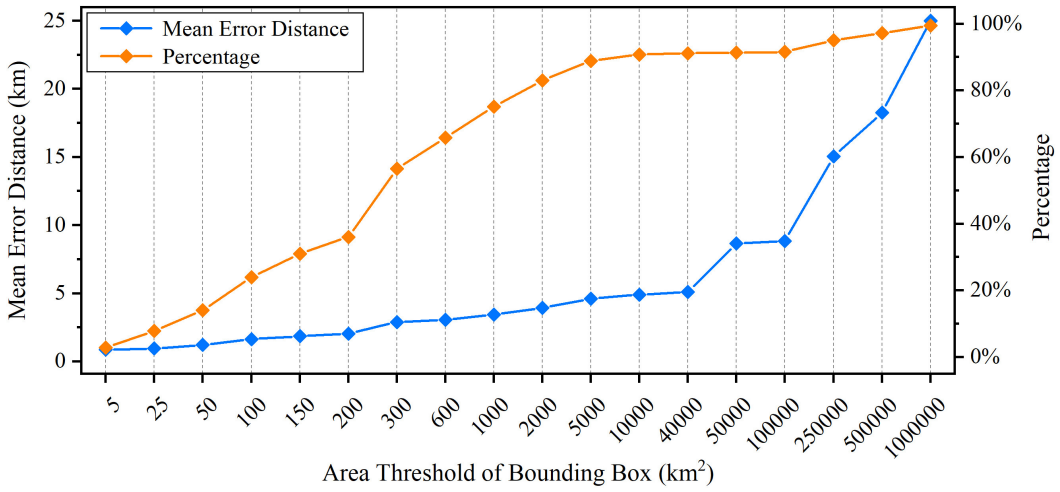


Figure 3.8: MED and percentage of different area thresholds.

more than 6 km has been kept. Figure 3.9 shows MED of TUPB in three datasets with different distance thresholds, when the area threshold is set to 5,000 km^2. As illustrated in this figure, the distance threshold has no obvious effect on datasets of DBC and GA, but it has a significant impact on DBA. When distance is set to 6 km, the MED is lowest, hence we chose 6 km as the distance threshold in this chapter.

3.4.3 Results

Combining models mentioned in Section 3.3, three coordinate datasets of counties of the US and Equation (3.6), MED ($B_{AREA_i} \leq 5,000$ km^2 and $B_{AREA_i} \leq 10,000$ km^2) can be computed and shown in Table 3.4 and Figure 3.10. When the area threshold is set to 5,000 km^2, about 88.9% of sample tweets are successfully predicted, and the percentage has improved to 90.8% when the area threshold is set to 10,000 km^2.

From Figure 3.10(a), we can see that GA has a relatively steady performance for all models, and all values of MED are around 4.62 km. DBC has a similar performance to GA, but the models with four sources have relatively worse performances com-

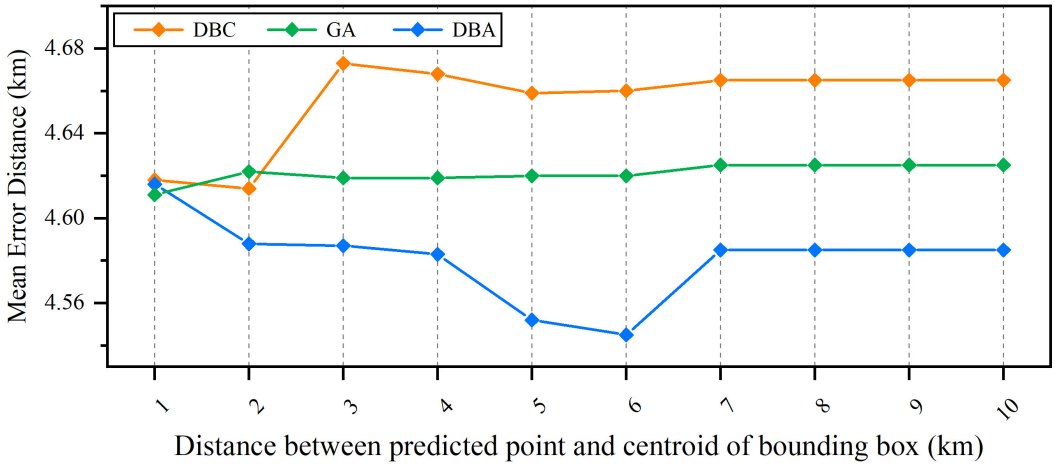


Figure 3.9: MED based on different distance thresholds.

pared to other models. While DBA has a clear trend of variation based on different models, the models with three or four sources have better performances than other models. Figure 3.10(b) shows MED's variation with respect to DBC, GA, and DBA based on 16 models when the area threshold of the bounding box is set to 10,000 km^2. We can see that three lines from Figure 3.10(b) have similar trend patterns as those from Figure 3.10(a).

There often exist some abnormal values in the dataset, and these values can pose a significant impact on the mean value, hence the median value can reduce the impact of abnormal values. Table 3.5 and Figure 3.11 show the median error distance with the bounding box's area of 5,000 km^2 and 10,000 km^2.

From Figure 3.11(a), we can see that the line of GA is almost straight, and all values are around 3.25 km. DBC shows a similar performance to GA, but three models of DBC performed relatively better. While DBA performs vary depending on different models, especially the models with four sources show better performances than other models. Figure 3.11(b) shows MDED's trend of DBC, GA, and DBA based on 16 models when the area threshold of the bounding box is set to 10,000 km^2. We can see that the models with four sources have the same performance regardless of DBC,

Table 3.4: MED of models based on two area thresholds.

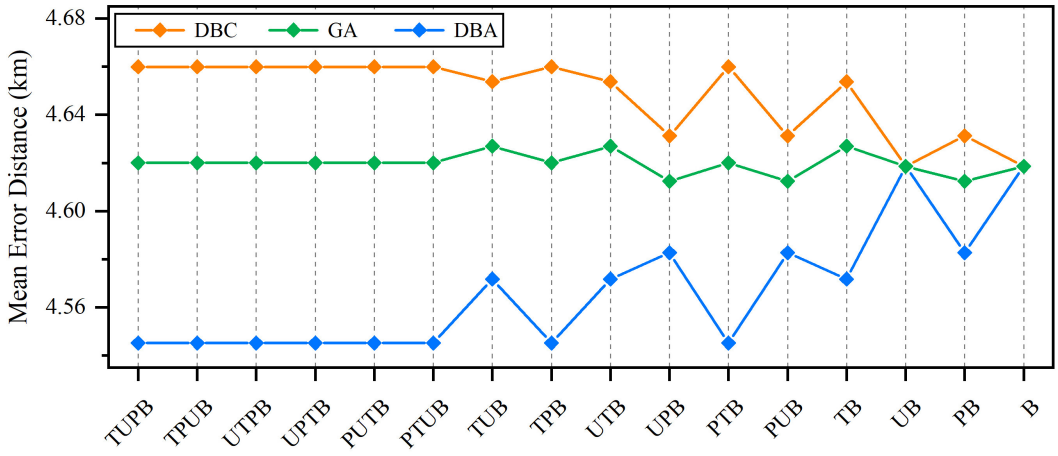
Models	MED ($B_{AREA_i} \leq 5,000$ km^2)			MED ($B_{AREA_i} \leq 10,000$ km^2)		
	DBC	DBA	GA	DBC	DBA	GA
TUPB	4.660	4.545	4.620	4.936	4.824	4.897
TPUB	4.660	4.545	4.620	4.936	4.824	4.897
UTPB	4.660	4.545	4.620	4.936	4.824	4.897
UPTB	4.660	4.545	4.620	4.936	4.824	4.897
PUTB	4.660	4.545	4.620	4.936	4.824	4.897
PTUB	4.660	4.545	4.620	4.936	4.824	4.897
TUB	4.654	4.572	4.627	4.930	4.850	4.904
TPB	4.660	4.545	4.620	4.936	4.824	4.897
UTB	4.654	4.572	4.627	4.930	4.850	4.904
UPB	4.631	4.583	4.612	4.908	4.860	4.890
PTB	4.660	4.545	4.620	4.936	4.824	4.897
PUB	4.631	4.583	4.612	4.908	4.860	4.890
TB	4.654	4.572	4.627	4.930	4.850	4.904
UB	4.619	4.619	4.619	4.896	4.896	4.896
PB	4.631	4.583	4.612	4.908	4.860	4.890
B	4.619	4.619	4.619	4.896	4.896	4.896

GA, and DBA. But the values of MDED change a lot when less than four sources are used.

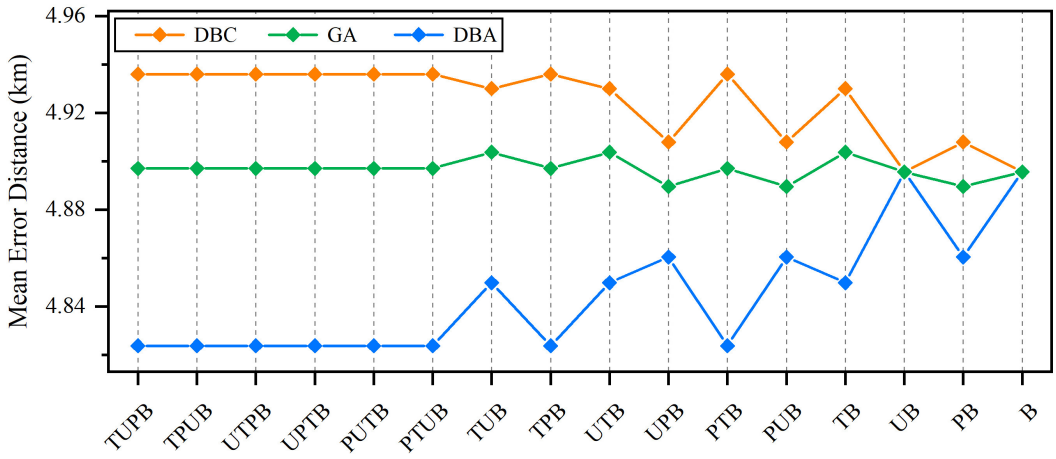
From Figure 3.10 (MED of the models) and Figure 3.11 (MDED of the models), it shows that DBA has the best performance in all cases, GA performs better in MED, and DBC performs better in MDED. Compared with MED, MDED have smaller error distances for all models.

Table 3.5: MDED of models based on two area thresholds.

Models	MDED ($B_{AREA_i} \leq 5,000$ km^2)			MDED ($B_{AREA_i} \leq 10,000$ km^2)		
	DBC	DBA	GA	DBC	DBA	GA
TUPB	3.239	3.095	3.245	3.327	3.233	3.373
TPUB	3.239	3.095	3.245	3.327	3.233	3.373
UTPB	3.239	3.095	3.245	3.327	3.233	3.373
UPTB	3.239	3.095	3.245	3.327	3.233	3.373
PUTB	3.239	3.095	3.245	3.327	3.233	3.373
PTUB	3.239	3.095	3.245	3.327	3.233	3.373
TUB	3.183	3.135	3.244	3.280	3.243	3.367
TPB	3.239	3.095	3.245	3.327	3.233	3.373
UTB	3.183	3.135	3.244	3.280	3.243	3.367
UPB	3.239	3.195	3.239	3.324	3.239	3.259
PTB	3.239	3.095	3.245	3.327	3.233	3.373
PUB	3.239	3.195	3.239	3.324	3.239	3.259
TB	3.183	3.135	3.244	3.280	3.243	3.367
UB	3.239	3.239	3.239	3.255	3.255	3.255
PB	3.239	3.195	3.239	3.324	3.239	3.259
B	3.239	3.239	3.239	3.255	3.255	3.255



(a) $B_{AREA_i} \leq 5,000$ km^2

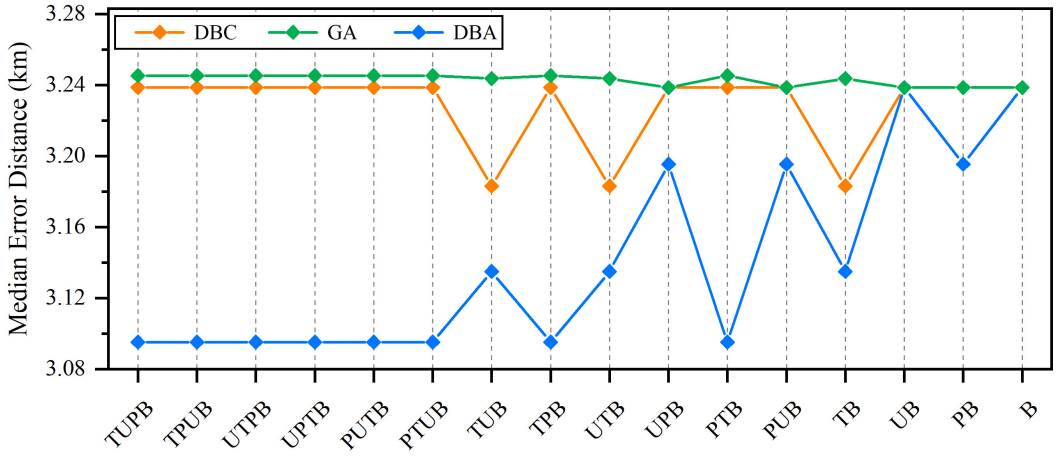


(b) $B_{AREA_i} \leq 10,000$ km^2

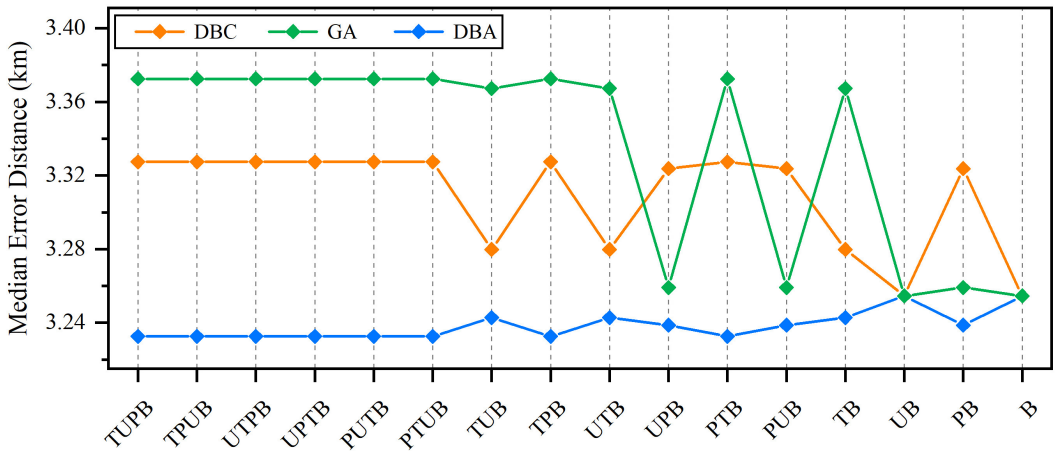
Figure 3.10: MED of models based on two area thresholds.

3.5 Chapter Summary

Twitter has demonstrated its importance for gathering and publishing up-to-date information during a real-world event. Geographic information plays an important role in emergency response and event monitoring. However, only 2% of tweets are with geo-tags, hence geolocation inference of tweets is still a major challenge. In this chapter, we proposed various models to predict geolocation of tweets, as organized as follows: (1) Twitter data collection, (2) data cleaning and extract geo-tagged tweets



(a) $B_{AREA_i} \leq 5,000$ km^2



(b) $B_{AREA_i} \leq 10,000$ km^2

Figure 3.11: MDED of models based on two area thresholds.

related to COVID-19, (3) location entity extraction from location-related metadata of tweets based on NER, (4) construction of three coordinate datasets on the basis of gazetteers and digital boundaries of the US, (5) model implementation based on different area thresholds of bounding box, and (6) model evaluation.

The proposed method has fully used all potential location-related attributes to predict tweets' geolocation. When the area threshold of the bounding box is set to 10,000 km^2, the best model can successfully predict the geolocation of 90.8% of COVID-19 related tweets with the mean error distance of 4.824 km and the me-

dian error distance of 3.233 km. This method has achieved the best performance compared with previous methods.

There still exist some deficiencies in this chapter. Firstly, the library of NER is limited and does not contain every county's name, which results in some useful information being filtered out. Secondly, even though the distance threshold is introduced to reduce the interference caused by duplicate county names, there still exist counties with the same name located in the same bounding box. Thirdly, several location entities can be extracted based on NER in some cases, but only the first location entity that meets the criteria is chosen in this chapter, even though there is a possibility that the real location-related information does not always appear in the first place. Our future work will focus on these limitations.

By following the data mining of Twitter dataset, a new data source of OFD data is introduced in the next chapter. This dataset can provide GPS coordinates and address information of every user, which is similar to the Twitter dataset. But we intend to extract the boundaries of Areas-Of-Interest (AOIs) based on an optimization model using OFD data.

Chapter 4

Simultaneous Detection of Multi-AOIs Using OFD Data

[2]In this chapter, a novel approach is proposed to detect multiple AOIs simultaneously and solve the multi-AOIs detection problem. In this approach, we first apply the existing single-AOI detection algorithms to generate candidate spatial boundaries for AOIs in a neighborhood, and then develop a Binary Integer Linear Programming (BILP) model to determine the best candidate spatial boundaries for these AOIs while accounting for their spatial dependency. We conduct numerical experiments using real data from Meituan, the largest OFD platform in China. Results show that our model not only produces consistent AOI boundaries but also improves the accuracy of multi-AOIs detection.

[2]Parts of this chapter have been published in Li, B., Chen, L., Xiong, D., Chen, S., He, R., Sun, Z., Lim, S., and Jiang, H. (2022). Simultaneous Detection of Multiple Areas-of-Interest Using Geospatial Data from an Online Food Delivery Platform. *ACM SIGSPATIAL '22: Proceedings of the 30th International Conference on Advances in Geographic Information Systems*, pp. 1-10. https://doi.org/10.1145/3557915.3561014.

4.1 Introduction

OFD platforms rely heavily on accurate AOIs information in their operations. An AOI, also known as a ROI, refers to a polygon selection in a map that someone may find useful or interesting, for example, a residential complex, a public park or a shopping mall [26, 27]. OFD platforms are concerned with two key properties associated with an AOI, that is, its name and spatial boundary. Spatial boundaries of AOIs are stored as vector formats, which can be easily used in geospatial analysis and improve service efficiency of the OFD industry. In Figure 4.1, we give an example of an AOI, whose name is *Yishashi Garden Apartments*, and the spatial boundary is coloured in blue. It is a gated apartment complex, and all 10 buildings of it are within the spatial boundary.

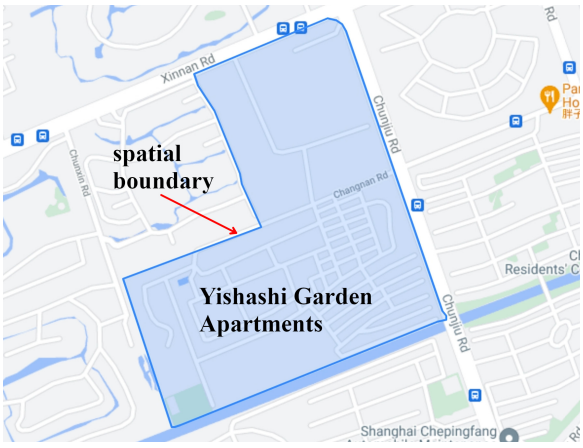


Figure 4.1: An example of an AOI.

An OFD platform uses AOI information in many ways, for example: (1) When a customer places an order, the OFD platform relies on AOI information to resolve the delivery address. Based on the GPS coordinates of customers and the boundaries of nearby AOIs, the platform would suggest possible AOI names to assist customers to pinpoint their exact locations, which is critical to ensure timely delivery. (2) On an OFD platform, a restaurant, say a McDonald's, is often asked to specify its

service area as a list of AOIs. The OFD platform then uses the GPS location of the customer together with the spatial boundaries of the AOIs to determine whether to show this McDonald's to the customer. If the OFD platform does not have the accurate spatial boundary for an AOI, for example, if the spatial boundary of *Yishashi Garden Apartments* is incorrect and Building #10 is erroneously excluded, the OFD platform would not allow residents living in Building #10 to order from this McDonald's, although residents living in other bulidings of this complex can. This would create inconsistent experience among residents living in the same gated complex.

OFD platforms have spent considerable efforts to improve the accuracy of their AOIs. Recently, they started to tap into the vast amount of geospatial data generated during the ordering process which is illustrated in Figure 4.2. When a customer takes out the phone to order, the OFD platform typically performs the following procedures.

- In Step 1, the OFD platform gets the GPS coordinates of the customer's location from the GPS chip of the phone, and by using the boundaries of known AOIs, it suggests possible AOI names as delivery addresses.
- In Step 2, if the customer does not modify the suggested AOI name, the AOI is proved to be accurate, otherwise, the modified address and GPS coordinates are sent to the geospatial database for potential AOIs.
- In Step 3, if the customer is not located in a known AOI, he/she is asked to type the address manually, which together with the GPS coordinates become the geospatial data that can be used to detect AOIs for potential AOIs.

In this chapter, we investigate how to detect AOIs from the geospatial data collected in the above process (see the red rectangle in Figure 4.2). The AOI detection

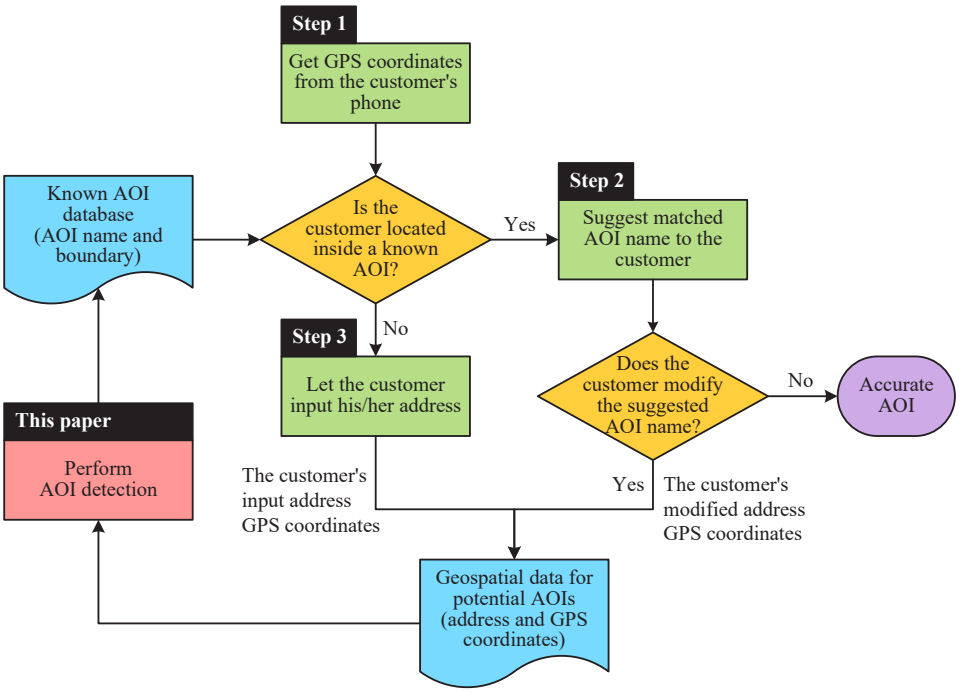


Figure 4.2: How AOI information is used in ordering process.

problem involves identifying the name and the boundary of an underlying AOI. Although there has been a proliferation of studies that investigate the AOI detection problem, existing approaches all focus on the single-AOI detection problem, that is, they detect AOIs one at a time. In fact, some noisy GPS points locate in the wrong locations, therefore, they may construct spatial boundaries of different AOIs with overlaps. Among existing studies, multiple geospatial data sources are applied into the single-AOI detection problem, including social media data (e.g., geo-tagged Flicker photos [28–30] and geo-tagged tweets [31–33]), remote sensing data [33], and delivery data [3].

Since single AOI detection algorithms detect AOIs independently of each other, they tend to produce inconsistent results. Take Figure 4.3 as an example. In Figure 4.3(a), there are two AOIs whose names are A and B. The blue dots are GPS locations located in AOI A, while the red dots are those located in AOI B. These GPS locations are generated directly from customers' mobile devices and collected

from the OFD platform. Existing studies would first use the blue dots to identify the spatial boundary of AOI A, which is shown in Figure 4.3(b), and then use the red dots to identify the spatial boundary of AOI B, which is shown in Figure 4.3(c). When we overlay the spatial boundaries of these two AOIs in Figure 4.3(d), they overlap with each other, which is inconsistent. In summary, different approaches of single-AOI detection tend to generate inconsistent results and cannot fully leverage GPS data in adjacent AOIs.

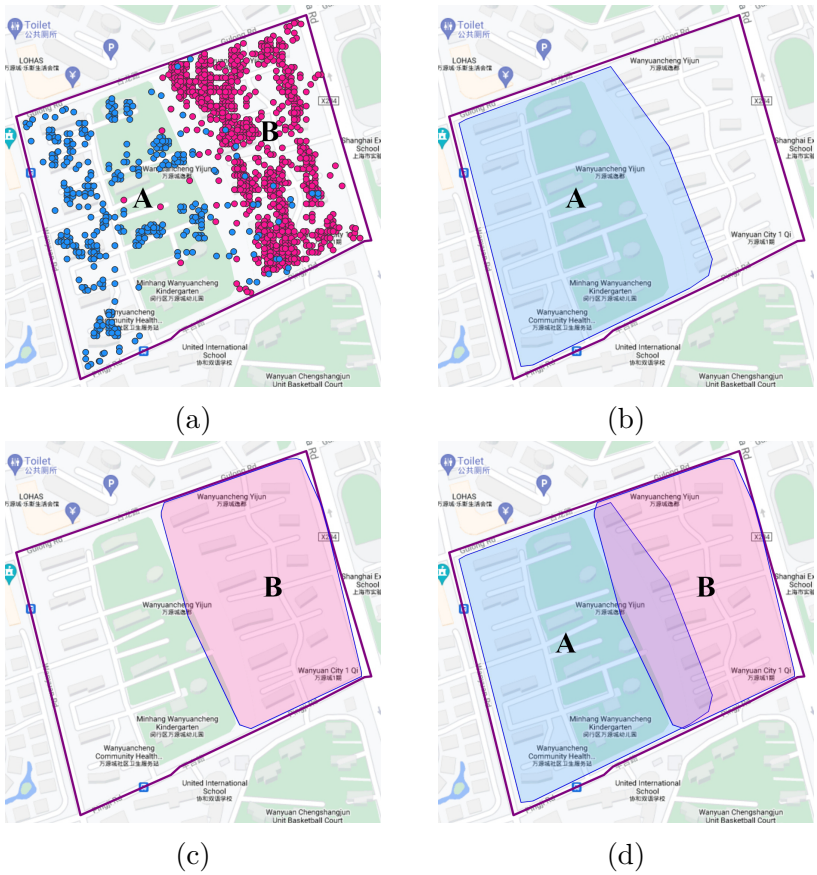


Figure 4.3: Existing studies detect AOIs independently of each other and may produce AOIs with overlaps. (a) shows the GPS locations for customers located in AOIs A and B. (b) and (c) illustrate the estimated spatial boundaries of AOIs A and B, respectively. (d) shows the spatial boundaries of AOIs A and B overlap, which is not acceptable.

In this chapter, we aim to address the challenge faced by single-AOI detection mod-

els through simultaneously detecting multiple AOIs. We propose to detect multiple AOIs simultaneously, that is, to solve the multi-AOIs detection problem. In our approach, we first apply existing single-AOI detection algorithms to generate candidate spatial boundaries for AOIs in a neighborhood, we then develop a BILP model to determine the best candidate spatial boundaries for these AOIs while accounting for their spatial dependency. We conduct numerical experiments using real data from Meituan, the largest OFD platform in China. Results show that our model not only produces consistent AOI boundaries but also improves the average F1-score by 4.7%. We improve the accuracy and preserve the details of the boundaries of detected AOIs by applying a Hidden Markov Model (HMM) to the road network dataset.

The contributions of this chapter can be summarized as follows.

- By accounting for the spatial dependency among neighbouring AOIs, we ensure that our approach can produce AOI boundaries that are consistent with each other.
- We formulate the problem as a BILP model, which can be efficiently solved by standard branch-and-bound procedures.
- Using the optimization model in the dataset collected from Meituan platform, results show that our model identifies Multi-AOIs and improves the average F1-score from 0.847 to 0.894 and achieves the best average F1-score among all single-AOI detection methods.

The rest of this chapter is organized as follows. Section 4.2 describes in detail of the optimization model. In Section 4.3, we perform numerical experiments to evaluate the benefits of our optimization model. Finally, we make a conclusion and describe the possible future expectations in Section 4.4.

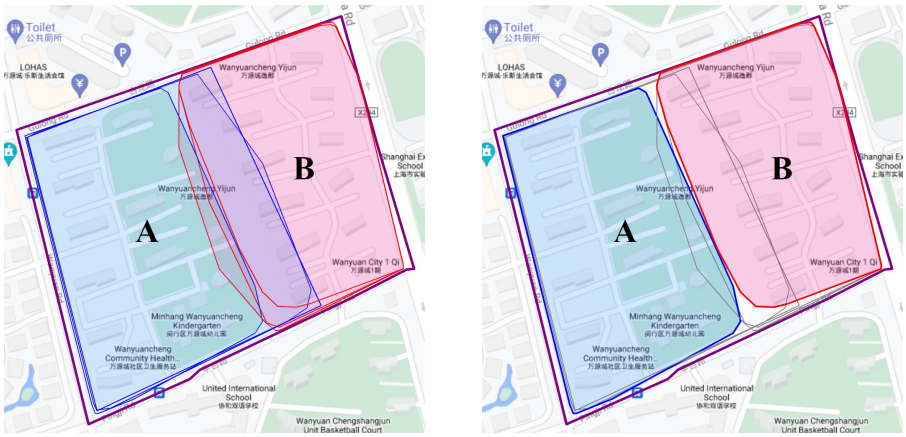
4.2 Methodology

We develop an optimization model for the simultaneous detection of multi-AOIs in this section. Inputs to this model include GPS points and multiple candidate spatial boundaries constructed by single-AOI detection models. The optimization model then outputs the optimal spatial boundaries of AOIs. We first use an example to illustrate how the model works, and then detail the optimization model. Finally, we refine the model by introducing the geohash technique.

4.2.1 The General Idea

For readers to understand our approach better, before diving into the details, we describe how the model works. We continue using the example mentioned in Section 4.1 to explain the working strategy of the optimization model. Figure 4.3(d) shows candidate spatial boundaries of AOIs A and B by a single-AOI detection algorithm. If we change the algorithms and their parameter values, different candidate spatial boundaries can be generated. For AOIs A and B, two sets of candidate sptial boundaries are generated as $\Psi_A = \{A_1, A_2, \cdots, A_n\}$ and $\Psi_B = \{B_1, B_2, \cdots, B_n\}$ by n single-AOI detection models. Then these data are fed into the optimization model.

Based on the mutual exclusion of different AOIs, the optimization model helps us discover the optimal spatial boundaries of AOIs A and B from Ψ_A and Ψ_B simultaneously. In this chapter, the definition of optimal spatial boundaries of AOIs is that spatial boundaries contain the corresponding GPS points as many as possible and there is no overlaps between any two of these spatial boundaries. Figure 4.4 illustrates the process of our proposed approach. In Figure 4.4(a), the blue and red lines represent candidate spatial boundaries of Ψ_A and Ψ_B, respectively. In Figure 4.4(b),



(a) Candidate spatial boundaries (b) Optimal spatial boundaries

Figure 4.4: Visualization of the proposed approach.

the optimal spatial boundaries of AOIs A and B are represented as the thick blue and red boundaries, and other candidate spatial boundaries are coloured in gray. The optimal spatial boundaries have no overlaps between each other and represent major regions of AOIs A and B.

Then the framework of the proposed approach is illustrated in Figure 4.5. First, geospatial data as inputs are fed into different single-AOI detection models, which are based on algorithms of G-ROI (detailed in Section 4.3.3.1), and DBSCAN (detailed in Section 4.3.3.2). Then different parameter values are assigned to single-AOI detection models to construct n models, which are labelled as Model-1, Model-2, $\cdots$, Model-n. Then these models are used to construct n candidate spatial boundaries of every AOI. Suppose there are m AOIs in the research region, and they are represented as AOI A, AOI B, $\cdots$, AOI M. From figure 4.5, we can see that $m \times n$ candidate spatial boundaries are constructed by different models. To make it easier to understand, we use different fill colours to represent candidate spatial boundaries constructed by different models. For example, candidate spatial boundaries created by Model-1 (i.e., $A_1, B_1, \cdots, M_1$) are filled with blue. And candidate spatial boundaries created by Model-2 and Model-n are filled with pink and green, respectively.

Then all of these candidate spatial boundaries are fed into the multi-AOIs detection model. After the process of the optimization model, finally, a set of optimized spatial boundaries of AOIs are discovered. For optimized spatial boundaries, different colours of AOIs are corresponded to different single AOI detection models.

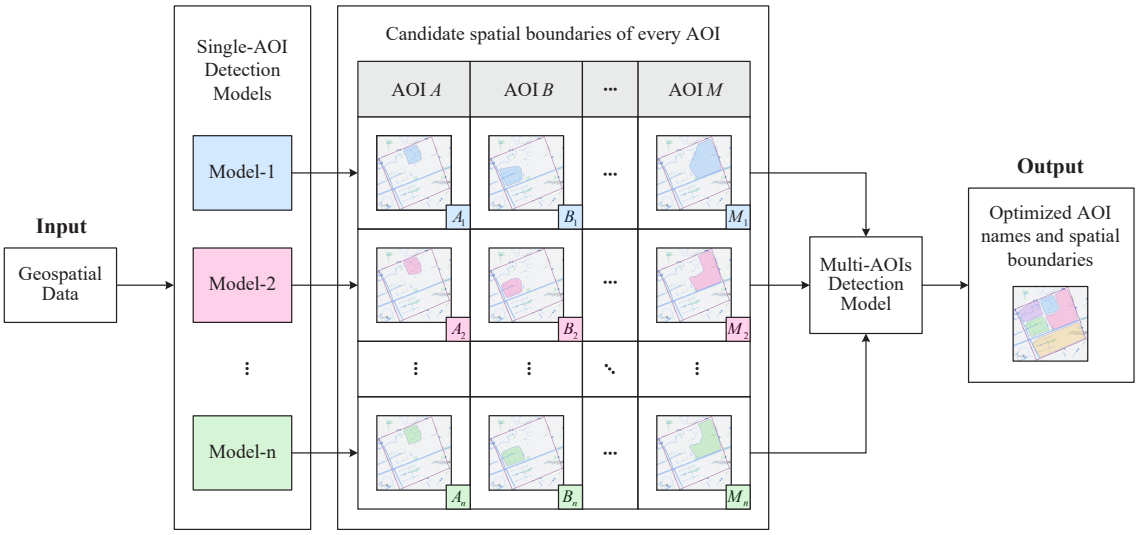


Figure 4.5: The Framework of the proposed approach.

4.2.2 Optimization Model

4.2.2.1 Notation and Terminology

Suppose A_i is one of the AOIs in our research area, where $i \in I$, and $C_{i,j}$ is one of the candidate spatial boundaries of A_i, where $j \in J$. Suppose P_k is one of the GPS points, where $k \in K$. And whether the GPS point P_k is located in $C_{i,j}$ or not is denoted by a binary variable $\delta_{i,j,k}$:

$$\delta_{i,j,k} = \begin{cases} 1, & \text{if the GPS point } P_k \text{ is located in } C_{i,j}; \\ 0, & \text{otherwise.} \end{cases} \tag{4.1}$$

Then we use a binary variable $\gamma_{k,i}$ to indicate whether the corresponding AOI name of P_k is A_i, if the answer is yes, then $\gamma_{k,i} = 1$, otherwise, $\gamma_{k,i} = 0$. That is,

$$\gamma_{k,i} = \begin{cases} 1, & \text{if } P_k\text{'s corresponding AOI name is } A_i; \\ 0, & \text{otherwise.} \end{cases} \tag{4.2}$$

If the candidate spatial boundary $C_{i,j}$ is chosen as the optimal spatial boundary of A_i by the optimization model, then $x_{i,j} = 1$, otherwise, $x_{i,j} = 0$. That is,

$$x_{i,j} = \begin{cases} 1, & \text{if } C_{i,j} \text{ is chosen as the optimal boundary;} \\ 0, & \text{otherwise.} \end{cases} \tag{4.3}$$

4.2.2.2 The Constraints

For any GPS point P_k and all candidate spatial boundaries containing it, at most one candidate spatial boundary can be chosen as the optimal spatial boundary. We apply this requirement by the following constraint:

$$\sum_{i \in I} \sum_{j \in J} \delta_{i,j,k} \cdot x_{i,j} \leq 1, \quad \forall k \in K. \tag{4.4}$$

For any AOI A_i, only one of the candidate spatial boundaries can be chosen as the optimal spatial boundary. This is achieved through the following constraints:

$$\sum_{j \in J} x_{i,j} = 1, \quad \forall i \in I. \tag{4.5}$$

$$x_{i,j} \in \{0, 1\}, \quad \forall i \in I, j \in J. \tag{4.6}$$

4.2.2.3 Model Formulation

The objective of this chapter can be regarded as a maximization problem. Thus, this problem is formulated as a BILP model, which is shown as follows:

$$\max \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} \delta_{i,j,k} \cdot \gamma_{k,i} \cdot x_{i,j}, \tag{4.7}$$

subject to constrains (4.4) through (4.6).

At last, the optimization model selects the optimal spatial boundaries for AOIs simultaneously.

4.2.3 Use Geohash to Improve Computational Performance

In Section 4.2.2, we have to set variables $\delta_{i,j,k}$ and $\gamma_{k,i}$ based on every GPS point P_k, but in fact many GPS points are located in the very close or even the same location. This fact can directly result in many GPS points being computed repeatedly and reducing computational efficiency in the optimization model. Therefore, we can make a grid with equal sized rectangular shaped cells with fine granularity, and every grid cell represents the whole GPS points located in it. For easier implementation of the grid, geohash technique is introduced in our study.

Geohash is a method of encoding geographic points into strings representing cells on the map. The size of a geohash cell is determined by a non-negative integer precision factor. In this research, we choose the precision factor with 9 and it can create square cells with 4.8m×4.8m. Then all GPS points located in the same geohash cells and with the same parsed AOI names are merged together, and represented as one geohash cell. And we also count the whole orders located in every geohash cell

as a weighted value. Finally, we use geohash cells to refine the modelling strategy.

Instead of setting variable $\delta_{i,j,k}$ based on P_k, we introduce G_l as one of the geohash cells, where $l \in L$. The centroid of a geohash cell means the geometric center of the cell. And whether the centroid of the geohash cell G_l is located in $C_{i,j}$ or not is denoted by a binary variable $\delta_{i,j,l}$:

$$\delta_{i,j,l} = \begin{cases} 1, & \text{if the centroid of } G_l \text{ is located in } C_{i,j}; \\ 0, & \text{otherwise.} \end{cases} \tag{4.8}$$

For the geohash cell G_l, we also count the total number of orders located in it as a weighted value w_l. This is denoted by a non-negative integer variable $\beta_{i,j,l}$:

$$\beta_{i,j,l} = \begin{cases} w_l, & \text{if the centorid of } G_l \text{ is located in } C_{i,j}; \\ 0, & \text{otherwise.} \end{cases} \tag{4.9}$$

Then we define a binary variable $\gamma_{l,i}$ to determine whether the corresponding AOI name of G_l is A_i, if the answer is yes, then $\gamma_{l,i} = 1$, otherwise, $\gamma_{l,i} = 0$. That is,

$$\gamma_{l,i} = \begin{cases} 1, & \text{if } G_l\text{'s corresponding AOI name is } A_i; \\ 0, & \text{otherwise.} \end{cases} \tag{4.10}$$

For any geohash cell G_l and all candidate spatial boundaries containing it, at most one candidate spatial boundary can be chosen as the optimal spatial boundary. We apply this requirement by the following constraint:

$$\sum_{i \in I} \sum_{j \in J} \delta_{i,j,l} \cdot x_{i,j} \leq 1, \quad \forall l \in L. \tag{4.11}$$

The definition of $x_{i,j}$ and constraints (4.5) and (4.6) remain unchanged. And the objective function is changed as follows:

$$\max \sum_{i \in I} \sum_{j \in J} \sum_{l \in L} \delta_{i,j,l} \cdot \beta_{i,j,l} \cdot \gamma_{l,i} \cdot x_{i,j}, \tag{4.12}$$

subject to constrains (4.5), (4.6) and (4.11).

4.2.4 Illustrative Example

For readers to understand the optimization model better, before diving into model details, we describe how the model works by an actual example. We chose a region in Shanghai with a boundary partitioned by the relatively higher road network. We collected delivery dataset with coordinates located in this region, and we got a dataset with 13,017 records.

For these records, they contain different kinds of noises. For example, they may contain some records of AOIs from the adjacent regions, and information of small POIs, like barber shops or restaurants, which should be filtered out from the dataset. In this chapter, we removed the useless records by the absolute quantity and density of orders. As for the absolute quantity of orders, we counted the number of orders corresponding to different AOI names, and deleted records with the number smaller than 80. Through this process, 563 records are deleted from the dataset.

For the rest of the records, actually, many of them are located in the very close place, which can increase the computational pressure of our model dramatically. Therefore, we introduce the geohash technique to cluster points that located in the close region as one bounding box. After using geohash to integrate coordinates located in the same cells, totally we got 1,525 geohash cells in this region. With

regard to the density of orders, we calculated the total number of orders located in every geohash cell, and deleted geohash cells with number of orders lower than 30. After this process, 88 geohash cells were deleted, then only 1,437 valid geohash cells were retained, which are represented as $\{G_0, G_1, \ldots, G_{1,436}\}$. And polygons were created by the valid geohash cells based on different parsed AOI names.

After data cleaning, we merged polygons of the same AOI. Generally, as for a specific AOI, it has different alias names and different forms of writing. For example, numbers can be typed into Chinese characters or Arabic numbers by customers. The other case is that customers may typed wrong Chinese characters. For these cases, we merged similar polygons based on rules as follows: (1) Text transformation: numbers of Chinese characters are converted into Arabic numbers. (2) Spatial similarity: detect alias names by the geometric properties of construed polygons, which is describe in Algorithm 1. After getting alias names, AOIs with alias names can be merged together. Through this process, we got 5 AOIs and set them as $\{A_0, A_1, A_2, A_3, A_4\}$.

After data pre-processing, we constructed 45 candidate spatial boundaries based on different single AOI detection algorithms (detailed in Section 4.3.3). Take the AOI A_0 as an example, it has 45 candidate spatial boundaries, which are represented as $\{C_{0,0}, C_{0,1}, \ldots, C_{0,44}\}$. For this case, totally, the number of all AOIs' candidate spatial boundaries is 5×45. Then we assigned values to the variables. For instance, if the geohash cell G_0 is located in the spatial boundary $C_{1,2}$, then we set $\delta_{1,2,0} = 1$, otherwise, $\delta_{1,2,0} = 0$. Follow the same rules, all variables should be set a value with 1 or 0, the total number is 5×45×1,437. At same time, we also record the number of corresponding orders located in the every geohash cell. For example, if $\delta_{1,2,0} = 1$, we count the number of orders located in the geohash cell G_0, and set the value to $\beta_{1,2,0}$, otherwise, $\beta_{1,2,0} = 0$. All the related variables should be set with a value, the total number is also 5×45×1,437.

Algorithm 1: Alias names Detection

Input: $polygonDict, p_1, p_2$

Output: $aliasNames$

```
1  begin
2      for (name_1, poly_1) ∈ polygonDict.items() do
3          for (name_2, poly_2) ∈ polygonDict.items() do
4              if name_1 ≠ name_2 then
5                  poly_1 ← Polygon(poly_1)
6                  poly_2 ← Polygon(poly_2)
7                  if poly_1 ∩ poly_2 ≠ φ then
8                      polyArea_1 ← area(poly_1)
9                      polyArea_2 ← area(poly_2)
10                     intersectArea ← area(poly_1 ∩ poly_2)
11                     unionArea ← area(poly_1 ∪ poly_2)
12                     r_1 ← intersectArea / unionArea
13                     r_2 ← intersectArea / polyArea_1
14                     r_3 ← intersectArea / polyArea_2
15                     if r_1 > p_1 or r_2 > p_2 or r_3 > p_2 then
16                         aliasNames.append([name_1, name_2])
17     return aliasNames
```

Then for any geohash cell, we have to compare its parsed AOI name with the AOI set $\{A_0, A_1, A_2, A_3, A_4\}$. For instance, if the parse AOI name of geohash cell G_2 is the same with the AOI A_4, then $\gamma_{2,4} = 1$, values of $\gamma_{2,0}$, $\gamma_{2,1}$, $\gamma_{2,2}$, and $\gamma_{2,3}$ are all set with 0. Follow the same rules, all variables should be set a value with 1 or 0, the total number is 5×1,437.

Apart from the known variables, we also have define the unknown variables, which are used to determine which spatial boundary is chosen for every AOI. For example, if the spatial boundary $C_{1,2}$ is the chosen optimal spatial boundary of AOI A_1, then the value of $x_{1,2}$ will be 1, otherwise the value will be 0. For every spatial boundary of each AOI, it has a variable to store this information, therefore, the total number of unknown variables is 5×45.

As for the constraints, we take G_0 as an example, maybe several candidate spatial boundaries contain G_0, but only one of them can be chosen as the optimal one. We apply this requirement by the constraint as:

$$\sum_{i=0}^{4}\sum_{j=0}^{44}\delta_{i,j,0}\cdot x_{i,j} \leq 1. \tag{4.13}$$

For any geohash cell G_k, it has a constraint like this, therefore, the total number of constraints for this requirement is 1,437.

Then for any AOI, we take A_1 as an example, only one of its candidate spatial boundaries can be chosen as the optimal one, therefore, only one of the values of $\{x_{1,0}, x_{1,1}, \ldots, x_{1,44}\}$ is 1, others are all 0. Thus we apply this requirement by constraints as:

$$\sum_{j=0}^{44} x_{1,j} = 1, \tag{4.14}$$

$$x_{1,j} \in \{0,1\}, \quad j = 0, 1, \ldots, 44. \tag{4.15}$$

For any AOI A_i, it has constraints like this, therefore, the total number of constraints for this requirement is 2×5.

The objective of this problem can be regarded as a maximization problem, and is shown as:

$$\max \sum_{i=0}^{4} \sum_{j=0}^{44} \sum_{k=0}^{1436} \delta_{i,j,k} \cdot \beta_{i,j,k} \cdot \gamma_{k,i} \cdot x_{i,j}, \tag{4.16}$$

subject to constrains (4.13) through (4.15).

Using the standard branch-and-bound algorithms to solve this BILP problem. We get the results with $x_{0,4} = 1$, $x_{1,21} = 1$, $x_{2,29} = 1$, $x_{3,0} = 1$, and $x_{4,30} = 1$, which means that $C_{0,4}$, $C_{1,21}$, $C_{2,29}$, $C_{3,0}$, and $C_{4,30}$ are the chosen optimal spatial boundaries of A_0, A_1, A_2, A_3, and A_4, respectively.

4.3 Numerical Experiments

To quantify the advantages of the proposed model, numerical experiments are performed. The computer programs for the optimization model are written in python and solved by IBM ILOG CPLEX 20.1.0 [151]. All computational tests are performed on a MacBook Pro equipped with an Intel 2.6 GHz CPU with 16 GB memory.

4.3.1 Dataset

Meituan, the offline-to-online (O2O) specialist, was founded in 2010, and now is one of the world's largest online food delivery platforms. It had 290 million monthly active users and around 600 million registered users as of April 2018 [152]. In this chapter, OFD data are collected from 4 Chinese cities for 3 months from October 1st to December 31st 2020 by Meituan platform. A description and example of

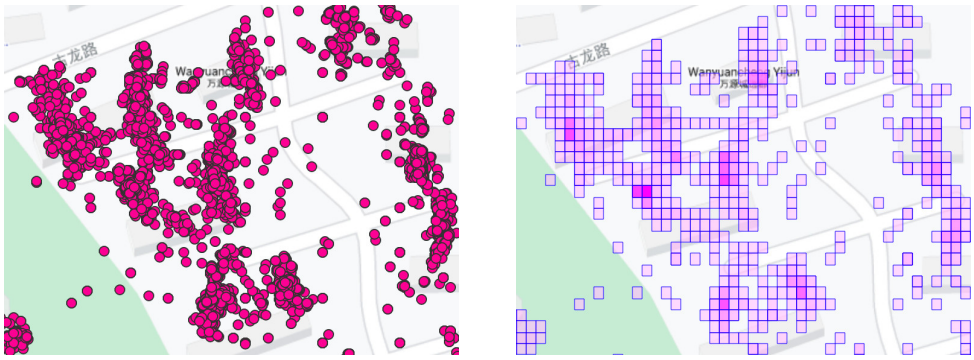
Table 4.1: Major data types of an order in Meituan.

Field	Description	Example
order_id	Unique identification of the order.	16015325886019701
create_dt	Created date and time of the order.	2020-11-05 12:03:31
user_id	Unique identification of the user.	320597876
user_lon	Longitude of the user's location.	121.397207
user_lat	Latitude of the user's location.	31.147946
user_addr	Text string of the user's address.	Room-N, XXX
rider_id	Unique identification of the rider.	14790342
delivery_dt	Completed date and time of the order.	2020-11-05 12:36:10

the dataset used in this chapter is shown in Table 4.1. For every order, it contains detailed GPS coordinates (fields of *user_lon* and *user_lat*) and address (the field of *user_addr*) of the user's location.

For the improved optimization model, before feeding the geospatial data into models, GPS points must be converted into geohash cells. We use an example to illustrate the process, which is shown in Figure 4.6. Figure 4.6(a) illustrates original GPS points of orders, and the corresponding geohash cells are shown in Figure 4.6(b). From Figure 4.6(b), the different colours of geohash cells indicate numbers of orders located in them, and the geohash cell with darker colour means it contains more orders. We treat every geohash cell as a GPS point, then feed them into single-AOI detection models.

Due to the limitation of variables and constraints of CPLEX, instead of implementing the optimization model throughout the whole city, we partition the city into multiple regions (also called large grids) based on road network and implement the model in every region separately. Road network data of these 4 cities are collected, and every road segment is stored as a polyline, which consists of a sequence of geographic coordinates. Every road segment has a property of road hierarchy, and the



(a) Original GPS points (b) Geohash cells

Figure 4.6: Visualization of geohash cells.

road network partition can be implemented by the road hierarchy, which is shown in Figure 4.7. Figure 4.7(a) illustrates road segments based on hierarchy 1−5. Then the road network help us partition the urban area into multiple regions, as shown in Figure 4.7(b) [153]. In this figure, we use different colours to distinguish these regions. For every region, the shape is around 1km×1km, and there are about 5 to 10 AOIs in it.

(a) Road segments (b) Hierachy-based region partition

Figure 4.7: Road network and small partitioned regions in Wangjing area of Beijing.

4.3.2 System Framework

The detailed system framework of the proposed approach is elaborated in Figure 4.8, consisting of four components.

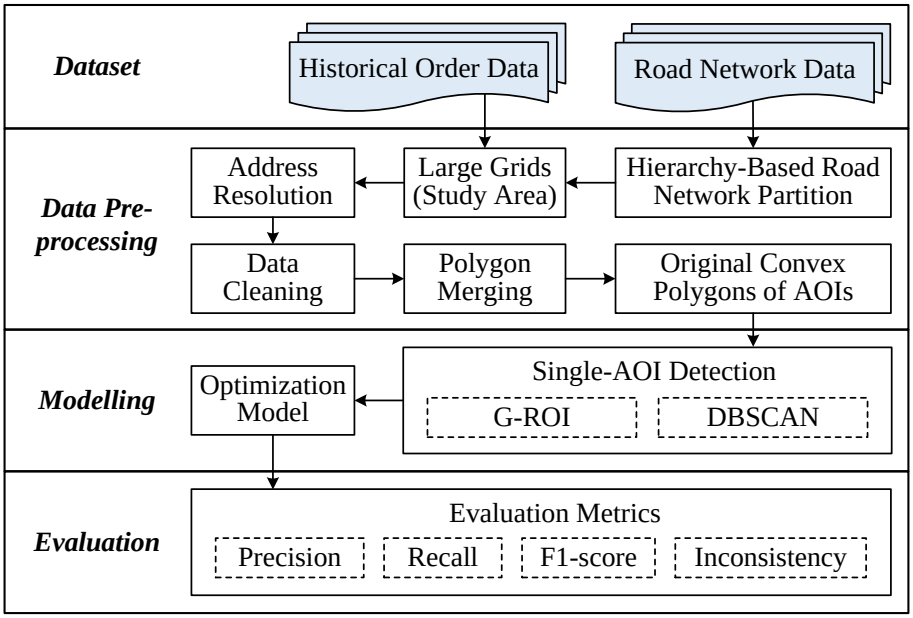


Figure 4.8: The detailed Framework of the proposed approach.

Dataset. This component includes two types of data: (1) *Historical Order Data*, which include detailed addresses and GPS coordinates of customers. (2) *Road Network Data*, which include detailed information of every road segment.

Data Pre-processing. This component takes dataset of historical order and road network, then performs 6 main tasks: (1) *Hierarchy-Based Road Network Partition*, which partitions the road network into large grids based on road hierarchy. (2) *Large Grids*, which are the partitioned regions and OFD data located in the specific large grid are captured. (3) *Address Resolution*, which extracts AOI names and geographic coordinates of every order in the large grid. (4) *Data Cleaning*, which removes outlier GPS points. (5) *Polygon Merging*, which merges the same AOI with alias names. (6) *Original Convex Polygons of AOIs*, which constructs original

boundaries of AOIs based on the convex hull.

Modelling. This component takes the preprocessed data as inputs, and generates boundaries of multi-AOIs, and then are fed into the optimization model. It includes two steps: (1) *Single-AOI Detection*, which generate a set of candidate spatial boundaries of every AOI based on algorithms of G-ROI and DBSCAN (detailed in Section 4.3.3). (2) *Optimization Model*, which uses the optimization model to detect multi-AOIs simultaneously based on the candidate spatial boundaries created in the previous step.

Evaluation. This component takes results of the optimization model, and evaluates the performance of results based on four metrics, which are precision, recall, F1-score and inconsistency (detailed in Section 4.3.5).

4.3.3 Baseline Algorithms

In this section, we briefly review the baseline algorithms used in our numerical experiments. According to existing studies in single-AOI detection, G-ROI algorithm [31] achieves the best F1-score compared with other methods, and DBSCAN algorithm [3, 28] has been widely used in AOI detection problems and achieves the robust performance. Additionally, we construct spatial boundaries of point clusters based on convex hull and modified alpha-shape concave hull.

4.3.3.1 G-ROI Algorithm

In [31], the authors propose a G-ROI algorithm for discovering ROIs on multiple social media datasets, and this algorithm achieves the best F1-score among other detection methods. This algorithm contains two stages of reduction and selection.

Let C be a set of geographic points within an AOI, and h_0 be the convex hull of C, and represented by a set of vertices.

Algorithm 2: Candidate spatial boundaries of G-ROI algorithm

Input: $geohashCells, perThresholds$

Output: $bounsGROI$

```
1  begin
2      for perThreshold ∈ perThresholds do
3          for aoi ∈ aois do
4              hulls, areas ← GROI_Reduce(geohashCells, aoi)
5              hull ← getHullbyPer(hulls, perThreshold)
6              bounConvex ← convexHull(hull)
7              bounConvexReg.append(bounConvex)
8          bounsGROI.append(bounConvexReg)
9      for aoi ∈ aois do
10         hulls, areas ← GROI_Reduc(geohashCells, aoi)
11         bounGROIConvex ← GROI_Selec(hulls, areas)
12         bounGROIConvexReg.append(bounGROIConvex)
13     bounsGROI.append(bounGROIConvexReg)
14     return bounsGROI
```

The reduction stage begins from h_0, then it finds one of its vertices to generate the smallest polygon and remove this vertex. With the same strategy, it continues until the convex hull containing only three vertices. This stage returns a set of convex hulls $H = \{h_0, h_1, \cdots, h_n\}$ and a set of removed points $P = \{p_0, p_1, \cdots, p_{n-1}\}$ obtained through the n steps that have been processed. The selection stage tries to discover the cut-off point p_{cut}. In this chapter, we set different parameter values to construct candidate spatial boundaries of AOIs, and then feed them into the

optimization model. Candidate spatial boundaries generated by G-ROI is described in Algorithm 2.

4.3.3.2 DBSCAN Algorithm

DBSCAN [69] is a density-based clustering algorithm, and is widely used in clustering for geospatial data. Compared with clustering methods like K-Means, DBSCAN does not require to specify the number of clusters, and can identify outlier points. Given a set of points on a Cartesian plane, DBSCAN can group together points with multiple nearby neighbours, and marks outlier points that lie alone in low-density regions (whose nearest neighbours are too far away). The working strategy behind DBSCAN is to identify the minimum number of neighboring points $minPts$ within the circle range of the radius ϵ. In comparison to other clustering algorithms, DBSCAN can identify clusters with different shapes and has good robustness for data noises [28].

Before using DBSCAN, the two parameters require to be set with proper values. The value of ϵ can be determined according to the geographic scale of the research problem. In general, if ϵ is set with a larger value, DBSCAN can construct AOIs with bigger coverage, while if the value is smaller, the produced AOIs are also smaller. The value of $minPts$ determines the minimum number of points of a cluster and represents the significance of the identified AOIs. If $minPts$ is set with a larger value, it can make sure to extract AOIs with a higher significance but may also miss some useful areas. while if $minPts$ is set with a smaller value, more clusters can be extracted but may also contain noisy points. In regard of the above reasons, it's difficult to find the best parameters of DBSCAN for every case, therefore, we set several groups of parameters to create different candidate spatial boundaries of AOIs, and discover the optimal one from the optimization model. Candidate spatial boundaries generated by DBSCAN is described in Algorithm 3.

Algorithm 3: Candidate spatial boundaries of DBSCAN

Input: $geohashCells, paramsList$

Output: $bounsDBSCAN$

```
1  begin
2      for (eps, minPts) ∈ paramsList do
3          for aoi ∈ aois do
4              cluster ← DBSCAN(eps, minPts, aoi)
5              bounConvex ← convexHull(cluster)
6              bounConcave ← bouncaveHull(cluster)
7              bounConvexReg.append(bounConvex)
8              bounConcaveReg.append(bounsConcave)
9          bounsDBSCAN.append(bounConvexReg)
10         bounsDBSCAN.append(bounConcaveReg)
11     return bounsDBSCAN
```

4.3.3.3 Concave Hull

After identifying clusters of customers' locations, the next step is to construct polygons from these GPS points. The convex hull is a typical way to represent the external polygon of points, and has been applied in many studies [28, 154]. While in some cases, the convex hull cannot match the boundary better but contains empty parts which do not belong to the original points. In our research, we try to find spatial boundaries of AOIs with no overlaps among each other, therefore, convex polygons cannot represent these AOIs properly. For more precise delineation of cluster shapes, [81] propose the algorithm of alpha-shape, which can be used to construct concave hulls and represents shapes of AOIs more properly.

The steps for concave hull computation can be summarized as follows: (1) Generate

a triangulated irregular network (TIN) of a set of discrete points using Delaunay triangulation method. (2) Remove exterior edges of the triangle with circumradius longer than a pre-defined length parameter l. (3) Repeat step 2 until every triangle's circumradius in the TIN is shorter than l. (4) Generate the resulted shape, which is the purple polygon in Figure 4.9(a). The parameter l can be equal to any positive number. If l is less than the shortest circumradius r_{min}, then all edges are removed. If l is bigger than the longest circumradius r_{max}, then all edges are kept, and the generated hull will be the convex hull of the points. Therefore, only values between r_{min} and r_{max} can be the optimal value.

(a) Original alpha-shape (b) Modified alpha-shape

Figure 4.9: Visualization of alpha-shape algorithms. (a) and (b) show the boundaries (purple) by the original and modified alpha-shape, respectively, and blue polygons are the deleted parts.

In this research, we modify the algorithm of alpha-shape to be more accurate for our dataset. For the algorithm of alpha-shape, it aims to delete all exterior edges with circumradius shorter than l, therefore, it will create some empty parts. For the modified alpha-shape, it can create smoother boundaries, as described in Algorithm 4. For the input of this algorithm, *points* means GPS points, and *per* refers to the

Algorithm 4: Modified alpha-shape

Input: $points, per$

Output: $concaveHull$

```
begin
    triangularMesh ← DelaunayTriangulation(points)
    for (i_a, i_b, i_c) ∈ triangularMesh.vertices do
        circumR ← circumradius(a, b, c)
        edgePointsAppend(edgePoints, points[[i_a, i_b, i_c]])
        circumRAppend(circumRList, points[[i_a, i_b, i_c]])
    circumRList ← sorted(circumRList)
    index ← round(per × len(circumRList)
    threshold ← circumRList[index]
    for i ← 0 to len(circumRList) do
        if circumRList[i] > threshold then
            deletedEdgePoints.append(edgePoints.pop(i))
    keptPolys ← cascaded_union(edgePoints)
    deletedPolys ← cascaded_union(deletedEdgePoints)
    concaveHull ← Polygon()
    if type(deletedPolys) = MultiPolygon then
        for poly ∈ deletedPolys do
            areaPolys.append(area(poly))
        index_1, area_1 ← getIndexArea(areaPolys, 1)
        index_2, area_2 ← getIndexArea(areaPolys, 2)
        if area_1/area_2 > 3.0 then
            deletedPoly ← deletedPolys.pop(index_1)
    concaveHull ← keptPolys ∪ deletedPolys
    return concaveHull
```

percentage of deleted triangles. We first calculate $triangularMesh$ by Delaunay triangulation method. Then, the cicumradius of every triangle is calculated and sorted from large to small. A threshold of circumradius is calculated by per, we keep all triangles with the circumradius shorter than the threshold. For other triangles, we use the method of *cascaded_union* to merge adjacent triangles together, therefore some separated polygons are created, which are blue polygons in Figure 4.9(a). We calculate areas of these polygons, and find the largest and second largest ones $area_1$ and $area_2$. If $area_1/area_2$ is larger than 3.0, only the largest polygons are deleted, which is the blue polygon in Figure 4.9(b), and other parts are merged together as the return concave hull, which is the purple polygon in Figure 4.9(b). Otherwise, the convex hull of *points* is returned.

4.3.4 Fine-tune Detected Boundaries

4.3.4.1 HMM Matching

In [155], the authors propose a map matching method by using HMM to discover the most probable route represented by a time series of GPS coordinates, and achieve a good performance. In this chapter, every candidate spatial boundary has vertices of GPS points, which can be applied to this algorithm. This algorithm can be used to fine-tune the detected boundaries to match road network. Since vertices of AOIs' actual boundaries are not necessarily positioned on the road network, therefore, it does not perform good enough in cases without high quality internal road network. In this chapter, we use two indices to evaluate results, which are defined as:

$$Ratio_1 = \frac{Area\left(AOI_{found}\right)}{Area\left(AOI_{found} \cup AOI_{HMM}\right)}, \tag{4.17}$$

$$Ratio_2 = \frac{Area\left(MRR\left(AOI_{found}\right)\right)}{Area\left(MRR\left(AOI_{found} \cup AOI_{HMM}\right)\right)}, \tag{4.18}$$

where AOI_{found} is the detected AOI's boundary. AOI_{HMM} is the boundary calculated by HMM matching. $MRR(\cdot)$ is a function to get the minimum rotated rectangle of a polygon. And only if both two ratios are larger than the threshold, the HMM matching result will be accepted, otherwise will keep the original detected AOI's boundaries.

4.3.4.2 Grid Matching

We partition every research region into small grids by the internal road network, then match small grids to AOIs. We determine which AOI do these grids belong to by the ratio defined as follows:

$$Ratio = \frac{Area\left(AOI_{found} \cap Polygon_{Grid}\right)}{Area\left(Polygon_{Grid}\right)}, \tag{4.19}$$

where $Polygon_{Grid}$ is the polygon of every small grid. Then, we merge all small grids belonging to the same AOI, and use strategies mentioned in Section 4.3.4.1 to determine whether to keep the result or not.

Finally, these two fine-tuning algorithms are combined together to fine-tune the detected spatial boundaries. We compare the two results of HMM matching and grid matching, and keep the one with bigger area as the final result of the combining result.

4.3.5 Performance Metrics

Metrics of precision and recall are used to evaluate the performance of baseline algorithms as well as our approach in detecting AOIs. The ground-truth AOIs in this chapter are manually labeled by ground survey. As in [31], let G_AOI_i be one of the ground-truth AOIs in a region, where $i \in I$, and let F_AOI_i be the corresponding found AOI by a single AOI detection method. Let $G\_AOI_i \cap F\_AOI_i$ be the corresponding true positive area, which is defined as the overlap of G_AOI_i and F_AOI_i. The two metrics are defined as:

$$\text{precision} = \frac{\sum_{i \in I} Area(G\_AOI_i \cap F\_AOI_i)}{\sum_{i \in I} Area(F\_AOI_i)}, \tag{4.20}$$

$$\text{recall} = \frac{\sum_{i \in I} Area(G\_AOI_i \cap F\_AOI_i)}{\sum_{i \in I} Area(G\_AOI_i)}, \tag{4.21}$$

where $\sum_{i \in I} Area(G\_AOI_i)$ is the area of all ground-truth AOIs, and $\sum_{i \in I} Area(F\_AOI_i)$ is the area of all found AOIs, and $\sum_{i \in I} Area(G\_AOI_i \cap F\_AOI_i)$ refers to the whole true positive area in the region.

To sort the results, the F1-score is defined as the harmonic average of the precision and recall as follows:

$$\text{F1-score} = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}. \tag{4.22}$$

To evaluate the overlap degree among different detected AOIs. We define the metric of inconsistency to calculate the overlap ratio of AOIs within a region as follows:

$$\text{Inconsistency} = \frac{Area\left(\cap_{i \in I} F\_AOI_i\right)}{Area\left(\cup_{i \in I} F\_AOI_i\right)}, \tag{4.23}$$

where $Area\left(\cap_{i\in I}F\_AOI_i\right)$ is the overlap area of all AOIs , and $Area\left(\cup_{i\in I}F\_AOI_i\right)$ is the union area of all AOIs in the region.

If the inconsistency is 0, it means all of those polygons are completely separated from each other. Higher inconsistency means a worse performance in this metric.

4.3.6 Optimization Results

In this chapter, we choose the densely populated area filled with large numbers of residential complexes as the research regions. Based on this principle, we choose 28 research cases from 4 Chinese cities. Two optimization models mentioned in Section 4.2 are represented as OM-1 (detailed in Section 4.2.2) and OM-2 (detailed in Section 4.2.3), which are applied to these research regions. After testing, two models achieve the same results on multi-AOIs detection, but they differ greatly in problem size. We report the summary statistics for OM-1 and OM-2 in Table 4.2. On average, they have the same number of binary variables, which is 199, but OM-1 has 13,017 constraints and OM-2 has only 1,525 constraints. We notice that instances can be solved in 78.7s and 13.8s by OM-1 and OM-2, respectively. Therefore, OM-2 has a significant efficiency improvement compared with OM-1.

Table 4.2: Optimization summary.

Design	Problem size		CPU time(s)
	Binary variables	Constraints	
OM-1	199	13,017	78.7
OM-2	199	1,525	13.8

Then we evaluate the average metrics of these research regions. Table 4.3 illustrates the performance (precision, recall, F1-score, and inconsistency) of all single-AOI detection models, and the best values of the whole models are shown on the last

row. G-ROI-01, G-ROI-02, $\cdots$, G-ROI-08 are models generated by G-ROI algorithm, and AVG-G-ROI refers to the average values of these 8 models. DBSCAN-01, DBSCAN-02, $\cdots$, DBSCAN-16 are models generated by DBSCAN algorithm, and AVG-DBSCAN represents the average values of these 16 models.

Results of the Table 4.3 show that the best precision of the whole single-AOI detection models is achieved by G-ROI, which is 0.983, while recall of it is the lowest one. The reason is that G-ROI algorithm removes too many points, therefore, in most cases, the ground-truth AOIs contain the found ones. DBSCAN achieves relatively high results (F1-score ranging from 0.850 to 0.869), and the best recall and F1-score of the whole single-AOI detection models is achieved by DBSCAN. On average of all models by DBSCAN, the precision is 0.895 and the recall is 0.825, which results in the F1-score of 0.858. The precision is bigger than the recall, which means the identified AOIs are smaller than the ground-truth ones.

Table 4.4 illustrates the evaluation results of convex hull, optimization model, and fine-tuning algorithms. For the original convex hull, the precision and recall are 0.827 and 0.865, respectively, which results in the F1-score of 0.847. The fact that the value of precision is lower than recall, means that the original convex hulls are on average bigger than the ground-truth ones. The inconsistency is 0.105, which means those AOIs have quite a lot of overlaps.

Then, the optimization model outperforms the other single-AOI detection algorithms in Table 4.3. The precision is 0.923 and the recall is 0.843, which leads to a F1-score of 0.881. These results support the ability of the optimization model to discover multi-AOIs simultaneously. The inconsistency is 0, which means that the spatial boundaries of different AOIs are completely separated.

Finally, algorithms of HMM and grid matching are applied to fine-tune the detected spatial boundaries of AOIs based on the road network data. After improved by

Table 4.3: Average Precision, Recall, F1-score, and Inconsistency of all single-AOI detection models in all cases.

Model	Precision	Recall	F1-score	Inconsistency
G-ROI-01	0.878	0.836	0.856	0.063
G-ROI-02	0.897	0.824	0.859	0.049
G-ROI-03	0.918	0.810	0.861	0.031
G-ROI-04	0.931	0.788	0.854	0.024
G-ROI-05	0.949	0.758	0.843	0.014
G-ROI-06	0.965	0.705	0.815	0.005
G-ROI-07	0.976	0.641	0.774	0.001
G-ROI-08	0.983	0.511	0.672	0.000
AVG-G-ROI	0.937	0.734	0.817	0.023
DBSCAN-01	0.851	0.865	0.858	0.080
DBSCAN-02	0.857	0.865	0.861	0.072
DBSCAN-03	0.887	0.851	0.869	0.038
DBSCAN-04	0.888	0.846	0.866	0.036
DBSCAN-05	0.889	0.844	0.866	0.036
DBSCAN-06	0.917	0.803	0.856	0.014
DBSCAN-07	0.918	0.801	0.856	0.014
DBSCAN-08	0.919	0.800	0.855	0.012
DBSCAN-09	0.858	0.838	0.847	0.048
DBSCAN-10	0.863	0.836	0.850	0.041
DBSCAN-11	0.900	0.837	0.867	0.022
DBSCAN-12	0.900	0.832	0.865	0.021
DBSCAN-13	0.901	0.830	0.864	0.021
DBSCAN-14	0.923	0.787	0.849	0.008
DBSCAN-15	0.924	0.786	0.850	0.008
DBSCAN-16	0.925	0.785	0.849	0.007
AVG-DBSCAN	0.895	0.825	0.858	0.030
Best values	0.983	0.865	0.869	0.000

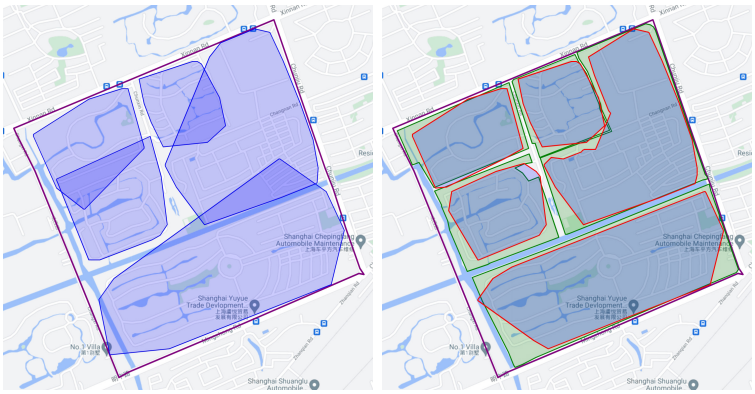
Table 4.4: Average Precision, Recall, F1-score, and Inconsistency achieved by convex hull, the optimization model, and fine-tuning algorithms in all cases.

Model	Precision	Recall	F1-score	Inconsistency
Convex Hull	0.827	0.868	0.847	0.105
Our Model	0.923	0.843	0.881	0.000
HMM Matching	0.888	0.896	0.892	0.000
Grid Matching	0.899	0.886	0.892	0.000
HMM+Grid Matching	0.892	0.895	0.894	0.000

the HMM algorithm, F1-score increases to 0.892, and this value is 0.892 based on the grid matching algorithm. For the algorithm combining both HMM and grid matching, the best F1-score is achieved, and the value is 0.894.

Figure 4.10 illustrates a sample diagram of the detected AOI boundaries based on the original convex hull method and our proposed model. Figure 4.10(a) shows the detected AOI boundaries based on the original convex hull method. From this figure, we can see that the overlaps between different AOI boundaries. Figure 4.10(b) illustrates the detected AOI boundaries based on our proposed model and ground truth AOI boundaries. In this figure, blue polygons represent the detected AOI, and green polygons represent the ground truth AOI. From this figure, we can see that no overlaps between different AOI boundaries.

Figure 4.11 visualizes the detected AOI boundaries of three fine-tuning algorithms and ground truth AOI boundaries. Figure 4.11(a) shows the detected AOI boundaries based on the HMM matching algorithm. Figure 4.11(b) shows the detected AOI boundaries based on the grid matching algorithm. Figure 4.11(c) shows the detected AOI boundaries based on the HMM+Grid matching algorithm. In these figures, blue polygons represent the detected AOI, and green polygons represent the ground truth AOI. From this figure, we can see that the detected AOI boundaries based on fine-tuning algorithms are better than the proposed model.



(a) Convex Hull (b) Our Model

Figure 4.10: Visualization of detected AOI boundaries. (a) Detected AOI boundaries based on the original convex hull method. (b) Detected AOI boundaries based on our proposed model and ground truth AOI boundaries.

(a) HMM Matching (b) Grid Matching (c) HMM+Grid Matching

Figure 4.11: Visualization of detected AOI boundaries of three fine-tuning algorithms and ground truth AOI boundaries.

4.4 Chapter Summary

In this chapter, a novel optimization model is proposed to detect multi-AOIs simultaneously. Results of numerical experiments suggest that the detection results are promising, and our model achieves the best average F1-score of the whole single-AOI detection models. To the best of our knowledge, this is the first instance of multi-AOIs detection. Moreover, we used two algorithms to fine-tune the detected spatial

boundaries based on road network and achieved better performance with respect to F1-score.

By following the detection of multi-AOIs using OFD data, the boundaries of residential complexes are extracted. Then the next task is to detect building footprints within residential complexes. In order to extract the detailed boundaries and names of building footprints, remote sensing images and OFD historical order data are introduced in the next chapter. A novel deep learning model is proposed to extract building footprints using remote sensing images.

Chapter 5

POI Detection of High-Rise Buildings Using Remote Sensing Images

[3]In this chapter, a multi-task Res-U-Net model with attention mechanism is developed for the extraction of the building roofs and the whole building shapes from remote sensing images, then use an offset vector method to detect the footprints of the high-rise buildings based on the boundaries of the corresponding building roofs and shapes. We also apply the OFD data to parse the POI name of every building footprint. Several strategies are also developed in combination with the proposed model, including data augmentation and post-processing. We conduct numerical experiments using real data of remote sensing images and OFD historical order

[3]Parts of this chapter have been published in Li, B., Gao, J., Chen, S., Lim, S., and Jiang, H. (2022). POI Detection of High-Rise Buildings Using Remote Sensing Images: A Semantic Segmentation Method Based on Multi-Task Attention Res-U-Net. *IEEE Transactions on Geoscience and Remote Sensing*, vol 60, pp. 1-16. https://doi.org/10.1109/TGRS.2022.3174399.

data. Results demonstrate that our proposed model achieves the best performance of F1-score and IoU in terms of both the building roof and shape segmentation.

5.1 Introduction

OFD platforms rely heavily on accurate Points-of-Interest (POIs) information in their operations. A POI is defined as a specific point location that may be useful or interesting for people, and a Region-of-Interest (ROI) is the POI's boundary [31], e.g., a residential building footprint. OFD platforms are concerned with two key properties associated with a POI, that is, its name and spatial boundary. Spatial boundaries of POIs are stored as vector formats, which can be easily used in geospatial analysis and improve service efficiency of the OFD industry.

In Figure 5.1, we give an example of POIs in a residential complex. In this figure, the dashed border represents the spatial boundary of an AOI, which is a gated residential complex. There are 30 buildings in the AOI, and gray polygons represent footprints of these buildings. For each building footprint, its centroid and building number are also shown in the figure. Each building is recognized as a POI, and its spatial boundary is the polygon of the building footprint and its name is the building number.

An OFD platform uses POI information in many ways, for example: (1) When a customer places an order, the OFD platform relies on the POI information to resolve the delivery address. Based on the GPS coordinates of customers and the boundaries of nearby POIs, the platform would suggest possible POI names to assist customers to pinpoint their exact locations, which is critical to ensure timely delivery. For example, a customer is located in the red point, which is shown in Figure 5.1, when he/she tries to choose the delivery address, the OFD platform can suggest relevant

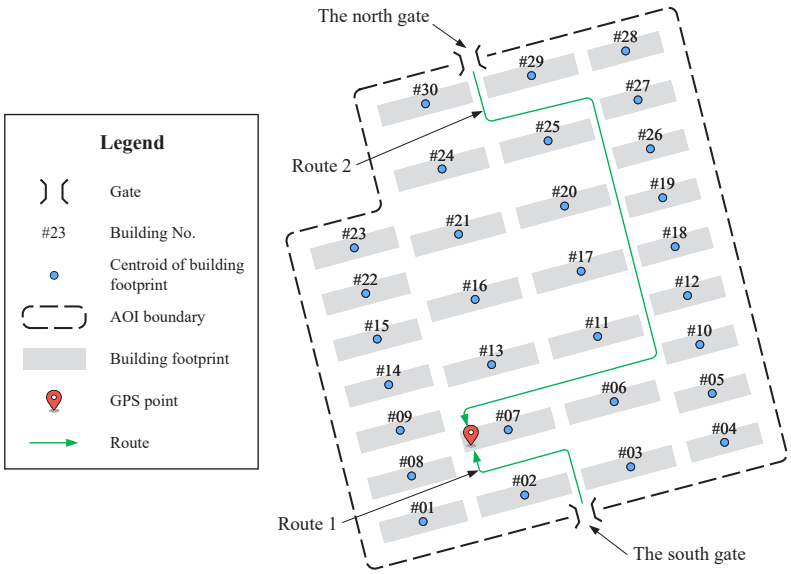


Figure 5.1: An example of POIs in a residential complex.

POI names in a list (e.g., Building #07, Building #08, Building #09, ...) based on the point's GPS coordinates, which can assist customers to find the right POIs of their delivery addresses. (2) The OFD platform can plan a more efficient route based on the accurate location, and riders, as known as food delivery men, can find the destination efficiently and accurately. For instance, the gated residential complex contains 30 buildings in Figure 5.1, if the OFD platform only plans a route to the gate of the residential complex, then riders have to find the destination building by their own familiarity with this place. The optimal route to the customer's place of the red point should be Route 1, while without accurate POI information, the OFD platform may plan a route to the wrong gate (i.e., the north gate) and riders may find the destination of Building #07 by Route 2, which is very inefficient and time-consuming.

ROI mining techniques are designed to detect boundaries of POIs. Existing ROI mining techniques are based on three primary approaches: predefined shapes, density-based clustering, and grid-based aggregation. Generally, the boundary of a POI is defined as a cluster, which is given by a geographically shaped convex polygon using

the convex hull of the geotagged records (e.g., geotagged Flickr photos) that include a given textual description such as a POI name within a circle to be optimized [2]. These existing methods highly rely on the accurate GPS coordinates, while the accuracy of indoor GPS satellite signals is relatively low, and high-rise buildings can also block or degrade GPS satellite signals of mobile devices, which may cause users to locate wrong locations. Therefore, these methods cannot extract POI boundaries accurately.

As remote sensing technologies mature, the capability to detect the physical boundaries of ground objects has been enhanced significantly. Thus, remote sensing technologies are seen as conventional and even important methods for detecting building footprints [34], which are also regarded as boundaries of POIs. In this chapter, we extract spatial boundaries of POIs based on remote sensing images. For example, different buildings are located in a residential complex, and every one of them can be recognized as a specific POI, and we aim to identify the spatial boundary and the name of each specific POI based on buildings from remote sensing images and OFD data.

Remote sensing images are now getting popular and broadly used in a variety of geoscience applications. Building extraction is advantageous for urbanization planning, disaster management, and environmental management [84, 156]. Because of the diversity of colours, sizes, shapes, and materials of buildings in different areas, and the similarity between buildings and backgrounds or other objects, developing accurate and robust extraction methods of buildings has become a challenging problem. For most existing studies, remote sensing data focus on low-rise buildings, and therefore building footprints can be generally extracted from the "footprint" of the roof [35–37]. However, for high-rise buildings, due to the different view angles of remote sensing sensors, the distance between the polygons of the building roof and the building footprint on remote sensing images can be very large and changes

with time. In this case, the polygon of the building roof cannot be regarded as the polygon of the building footprint. Besides building footprints cannot be extracted directly, because most of them are covered by building roofs and building surfaces on remote sensing images. Based on data analysis and calculation, we found that, in most cases, the building footprint has the same shape as the building roof, except for a certain offset in position. Therefore, we design a multi-task deep learning models to extract boundaries of the building roofs and the whole building shapes, then an offset vector method is applied to boundaries of the building roofs, which can help to get boundaries of the building footprints.

In this chapter, we aim to tackle the challenge faced by semantic segmentation of building footprints for high-rise buildings based on remote sensing images. We propose a multi-task Res-U-Net model with attention mechanism to extract the boundaries of building roofs and shapes simultaneously, and then use an offset vector method to detect the actual spatial boundaries of the building footprints based on the spatial boundaries of the building roofs and shapes. After detecting the spatial boundaries of the building footprints, parsing of POI names is processed by the OFD dataset. We conduct numerical experiments using remote sensing data from Google Earth and the OFD dataset from Meituan platform. Experimental results indicate that the proposed method successfully extracts the boundaries of the building roofs and shapes, and improves the total F1-score by 1.78% and 3.31% for the building roof segmentation and the whole building shape segmentation, respectively. The offset vector method helps to obtain the spatial boundaries of building footprints, and OFD dataset helps to parse POI names.

In this chapter, we propose a multi-task Res-U-Net model with attention mechanism to extract both the building roofs and shapes of high-rise buildings from remote sensing images. Our proposed model obtains an overall F1-score of 77.05% and IoU of 63.55% in terms of the building roof segmentation, and an overall F1-score of

79.02% and IoU of 66.05% for the whole building shape segmentation, which both achieve the best performance among baseline models.

The main contributions of this chapter can be summed up as follows.

- We propose a novel multi-task Res-U-Net model with attention mechanism for semantic segmentation of the building roofs and shapes. Using the proposed model, the building roofs and the whole building shapes are extracted simultaneously. Even compared with the best performing baseline model, the proposed model improves the total F1-score by 1.78% and IoU by 0.49% in terms of the building roof segmentation, and the total F1-score by 3.31% and IoU by 3.03% for the whole building shape segmentation.
- Most of existing studies extract building roofs as building footprints, while in our study, we introduce an offset vector method to extract the building footprints based on boundaries of the building roofs and the whole building shapes for high-rise buildings.
- Instead of detecting spatial boundaries of the building footprints, our research also parses POI names based on the OFD dataset.

The rest of the chapter is arranged as follows. Section 5.2 introduces the architecture of the multi-task Res-U-Net model with attention mechanism, the offset vector method, and the name parsing of POIs. In Section 5.3, we perform numerical experiments to describe the experimental results of our proposed model. We discuss and analyze the extraction results of the building roofs and the whole building shapes based on different models and the ablation study in Section 5.4. Section 5.5 sums up the conclusions of this chapter and describes the possible future expectations.

5.2 Methodology

This section will detail the proposed multi-task Res-U-Net model with attention mechanism for remote sensing images segmentation. We first describe the regular U-Net model, and two enhanced components including the ResNet block and the attention mechanism, then describe the overall structure of our proposed model and post-processing, finally, followed by the offset vector method and name parsing of POIs.

5.2.1 U-Net

In [97], the authors proposed the U-Net model, which is essentially a variant of FCN. U-Net is a type of deep CNN models containing connected convolutional layers and deconvolutional layers for segmentation of bio-medical images. U-Net has a symmetrical structure of the left encoder and the right decoder. For the left side, spatial features are extracted based on inputs. For the right side, segmentation maps are built from the extracted spatial features. And the decoder part is designed to build the segmentation map based on the extracted spatial features. The decoder is similar in structure of FCN using the combination of multiple convolutional layers. More precisely, the encoder is constitutive of a series of blocks with operations of down-sampling, and every block consists of repeated operations of two 3×3 convolutions and an operation of 2×2 max-pooling with stride 2. After each operation of down-sampling, the amount of filters of the convolutional layers will be doubled. Finally, the encoder part uses operations of two 3×3 convolutions to connect with the decoder part.

On the contrary, the decoder includes a series of blocks for operations of up-sampling, which are used to build segmentation images. The decoder uses an operation of 2×2

deconvolution to up-sample the feature map, which is produced by the encoder. In [157], the authors proposed the deconvolutional layer, which includes the operation of transposed convolution, and halves the amount of output filters. And followed by a series of blocks for operations of up-sampling, which contain two operations of 3×3 convolution and an operation of deconvolution. Then, the last layer is a 1×1 convolutional layer, which is used to output the segmentation results. The activation function of the output layer is the Sigmoid function, and all other layers adopt Rectified Linear Unit (ReLU) function. Definition of these two activation functions are as follows:

$$\text{Sigmoid function: } f(x) = \frac{1}{1+e^{-x}}, \tag{5.1}$$

$$\text{ReLU function: } f(x) = \max(0, x). \tag{5.2}$$

5.2.2 Residual Network

Residual Network (ResNet) [158] is a branch of DNN, and proposed to address the problem of degradation in optimization. To address such a problem, He *et al.* [158] proposed a deep residual learning framework where sets of layers fit a residual map, in contrast with other architectures which hope that each layer fits the entire underlying map.

Let $H(x)$ represents the last desired mapping, then,

$$F(x) = H(x) - x, \tag{5.3}$$

where x denotes the input image, and $F(x)$ represents the residual mapping (the

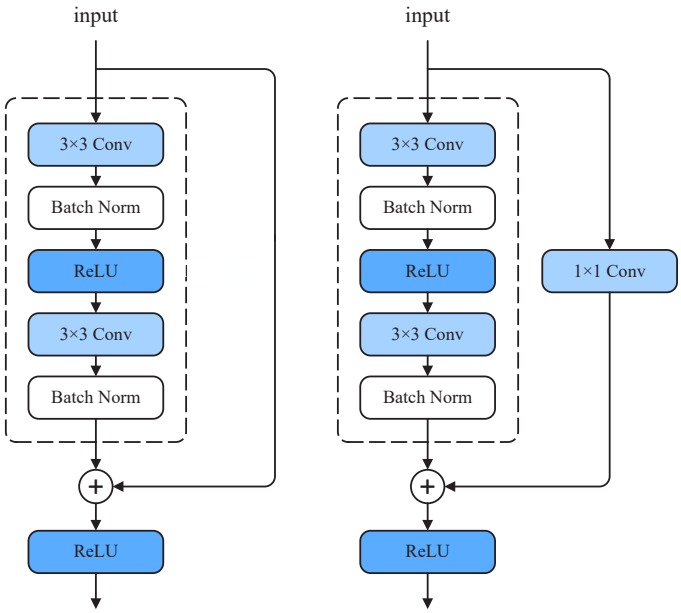


Figure 5.2: ResNet block (left: identity shortcut; right: projection shortcut).

portion within the dotted-line box in Figure 5.2). In a feedforward DNN, Equation (5.3) is called as a shortcut connection. More specifically, Equation (5.3) can be rewritten as:

$$y_i = F(x_i, \{W_i\}) + x_i, \tag{5.4}$$

where y_i and x_i stand for, respectively, the output and input vectors of the ith set of stacked layers, and $F(x_i, \{W_i\})$ is the shortcut connection to be learned. Because the shortcuts should have the equal size with x, we can map these as *identity* or as a linear projection through short connections to match the dimensions. Figure 5.2 illustrates an instance of ResNet block, which is utilized to construct the network. In the ResNet block, every convolutional layer is connected to a batch normalization [159]. In our proposed model, the number of channels of x_i should be changed in the ResNet block, therefore, the ResNet block with projection shortcut is applied to our model.

5.2.3 Attention Gate

Attention Gate (AG) for image analysis was introduced by Oktay *et al.* [160]. AG can retrieve information with high dimensions, and exclude information of backgrounds through the attention mechanism, which is illustrated in Figure 5.3. The convolutional layer provides a representation of images with high dimensions (x^l) by layer of local information. In the end, pixels are separated based on semantic differences of target features in space with high dimensions. The feature map is acquired through a linear conversion and the ReLU function at the output of the layer l, and the ReLU function can be represented as: $\sigma_1(x^l_{i,c}) = \max(0, x^l_{i,c})$, where i and c denote spatial and channel dimensions, respectively. Then followed by an operation of 1×1 convolution with just 1 filter and the activation function of Sigmoid. Based on this process, all values of the feature map are scaled to the interval [0, 1].

Attention coefficients, $\alpha_i \in [0, 1]$, identify prominent image areas and mitigate the feature response to keep activations related to the particular task. The result of attention gates are calculated as: $\hat{x}^l_{i,c} = x^l_{i,c} \cdot \alpha^l_i$. A unique scale attention value is calculated for every vector of pixels $x^l_i \in \mathbb{R}^{F_l}$, where F_l represents the amount of corresponding feature maps of layer l. As shown in Figure 5.3, a gating vector $g_i \in \mathbb{R}^{F_g}$ is utilized for every pixel i to identify focus areas. The gating vector includes contextual information to mitigate responses of lower-level features [161].

5.2.4 Model Architecture

The architecture of multi-task Res-U-Net model with attention mechanism is an encoder-decoder network on the base of U-Net structure, with a shared encoder path to the left and two structurally similar decoder path to the right, namely (1) *Shared Encoder*, (2) *Building Roof Decoder*, and (3) *Building Shape Decoder*. The

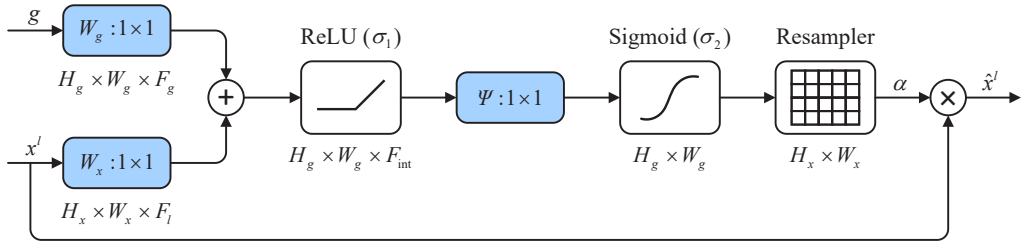


Figure 5.3: Schematic diagram of attention gate. Input features (x^l) are scaled based on attention coefficient (α) calculated from AG. Spatial areas are chosen by analyzing activations together with contextual information from the gating signal (g). Grid re-sampling of attention coefficient is used to make it have the same height and width with x^l.

outline of the proposed model is illustrated in Figure 5.4.

(1) Shared Encoder. The encoder is utilized to encode features on different levels, and then shared with two decoders. More precisely, the encoder path contains repeated operations of down-sampling which double the amount of feature channels and halve the size of feature maps in every step. Every operation of down-sampling is preceded by a ResNet block, which is mentioned in Section 5.2.2.

(2) Building Roof Decoder. This is to predict the segmentation of the building roofs from information of the shared encoder (i.e., the building roof decoder task). More precisely, every decoder block is symmetrical to the encoder, and in every decoder layer, features of the corresponding encoder layer are concatenated with an attention module, which enables the maintenance of multi-scale features, the enhancement of important channel features and the weakening of unimportant channel features.

(3) Building Shape Decoder. This is to predict the segmentation of the whole building shapes based on the information of the shared encoder and the building roof decoder. This is similar to the building roof decoder except that when merging the information in every layer, we concatenate not only the information of the shared encoder but also the information of the building roof decoder. The building

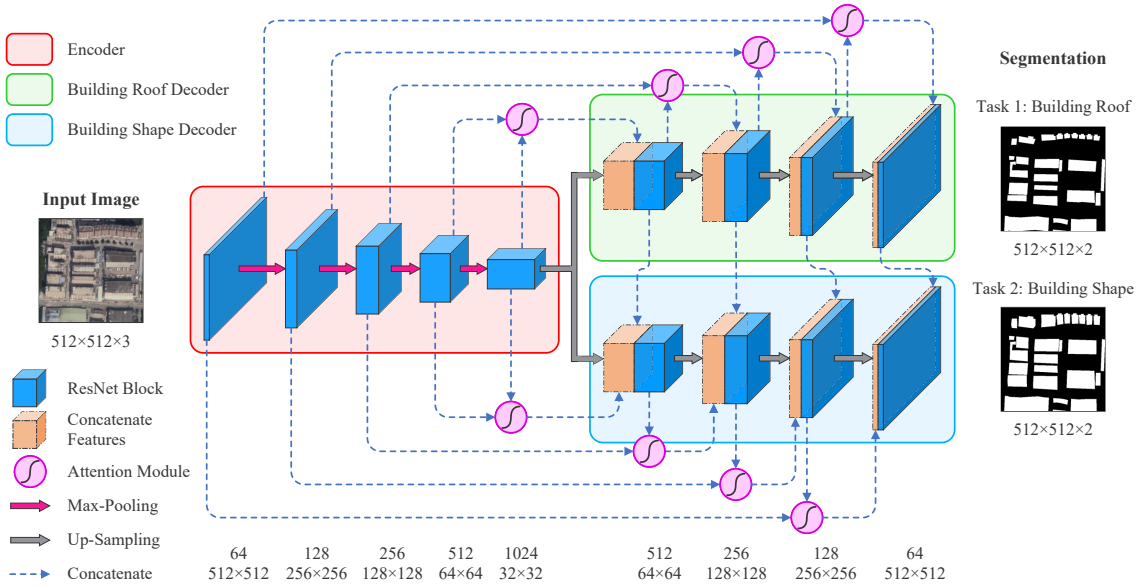


Figure 5.4: Architecture of the proposed multi-task Res-U-Net model with attention mechanism. Every blue cuboid represents a ResNet block. The number of channels and x-y-size are denoted on the bottom of the figure. The attention module filters the features propagated via the skip connections and dotted orange cuboids represent concatenate features. The arrows denote the different operations.

roof is always a part of the whole building shape, hence the learned building roof information would help with the segmentation prediction task of the whole building shape.

5.2.5 BCE and Dice Loss

Loss functions define how models calculate the overall error of predicted results and ground truth. The choice of loss functions can directly affect the learning process and the results of models. Even a properly constructed model can be affected by an improper loss function. In this research, extraction of the building roof and the whole building shape can be defined as two problems of binary segmentation in which the loss function of Binary Cross-Entropy (BCE) is often used, as indicated by Equation (5.5). However, remote sensing images of buildings exist an unbalanced

issue between pixels of buildings and backgrounds in which the loss function of BCE tends to be trapped at the local minimum, and the network is prone to predict a good loss value of the background and does not learn the feature representation of the minor class. One solution is adding weights to these classes during the loss calculation, which introduces extra hyperparameters that require careful adjustment. The other way is to select a more balanced function, e.g., BCE-Dice loss.

The definition of BCE-Dice loss is the summation of BCE loss and Dice loss, which combines the benefits of both functions. The BCE loss can express the erroneous classification well, and is easy to calculate the gradient in spite of the defect mentioned above. The Dice loss function, as indicated by Equation (5.6), is constructed on a Dice coefficient, which measures the overlapping area of the predicted result and the ground truth. If they match each other more, the Dice coefficient is closer to 1, driving the value of Dice loss closer to 0. On the contrary, if every pixel is predicted with the wrong value, the largest value of Dice loss is 1. However, BCE loss performs better in this point, as the loss function of BCE can generate values far greater than 1, which speeds up the progress of optimization. The BCE-Dice loss is indicated by Equation (5.7).

The prediction and ground truth of the building roof are denoted as $\hat{\mathbf{Y}}_r \in \mathbb{Z}_2^{I \times J}$ and $\mathbf{Y}_r \in \mathbb{Z}_2^{I \times J}$, respectively, where $\mathbb{Z}_2$ means the set of binary values $\{0, 1\}$. The prediction and ground truth of the whole building shape are denoted as $\hat{\mathbf{Y}}_s \in \mathbb{Z}_2^{I \times J}$ and $\mathbf{Y}_s \in \mathbb{Z}_2^{I \times J}$, respectively. Specifically, the BCE loss of the prediction $\hat{\mathbf{Y}}$ and the ground truth $\mathbf{Y}$ is:

$$\mathcal{L}_{BCE}(\hat{\mathbf{Y}}, \mathbf{Y}) = -\sum_i^I \sum_j^J \left[Y_{ij} \cdot log(\hat{Y}_{ij}) + (1 - Y_{ij}) \cdot log(1 - \hat{Y}_{ij}) \right], \tag{5.5}$$

where I and J are pixel numbers in height and width directions, respectively, i and j are the pixel indices, $Y_{ij} \in \{0, 1\}$ is the pixel label class, $\hat{Y}_{ij} \in [0, 1]$ is the probability

of the pixel being predicted as the positive category. In this research, pixels of the building roof and the whole building shape are the positive category, and pixels of the background are negative category.

The Dice loss of the prediction $\hat{\mathbf{Y}}$ and the ground truth $\mathbf{Y}$ is:

$$\mathcal{L}_{Dice}(\hat{\mathbf{Y}}, \mathbf{Y}) = 1 - \frac{2 \cdot \sum_i^I \sum_j^J \left(\hat{Y}_{ij} \cdot Y_{ij}\right) + \epsilon}{\sum_i^I \sum_j^J \hat{Y}_{ij} + \sum_i^I \sum_j^J Y_{ij} + \epsilon}, \tag{5.6}$$

where ϵ is utilized to guarantee the stability of the loss function by avoiding the numeric problem of being divided by 0.

The BCE-Dice loss of the prediction $\hat{\mathbf{Y}}$ and the ground truth $\mathbf{Y}$ is:

$$\mathcal{L}_{BCE-Dice}(\hat{\mathbf{Y}}, \mathbf{Y}) = (1 - \lambda) \cdot \mathcal{L}_{BCE}(\hat{\mathbf{Y}}, \mathbf{Y}) + \lambda \cdot \mathcal{L}_{Dice}(\hat{\mathbf{Y}}, \mathbf{Y}), \tag{5.7}$$

where λ denotes the hyper-parameter that balances BCE loss and Dice loss. If the boundary information of the building roof and the whole building shape are more complex, then the value of λ needs to be decreased; in contrast, if the boundary information of the building roof and the whole building shape are more regular, then the value of λ needs to be increased [109]. In our research area, the boundary information of the building roof and the whole building shape are more regular, and the value of λ is set to 0.8.

We minimize the hybrid loss of the building roof prediction and the whole building shape prediction by gradient descent.

$$\mathcal{L}(\boldsymbol{\theta}) = (1 - \varphi) \cdot \mathcal{L}_{BCE-Dice}(\hat{\mathbf{Y}}_r, \mathbf{Y}_r) + \varphi \cdot \mathcal{L}_{BCE-Dice}(\hat{\mathbf{Y}}_s, \mathbf{Y}_s), \tag{5.8}$$

where $\boldsymbol{\theta}$ represents a collection of all trainable parameters and φ denotes the hyper-parameter that balances the hybrid loss of the building roof prediction and the whole building shape prediction. If the boundary information of the building roof is more important than that of the whole building shape, then the value of φ needs to be decreased; in contrast, if the boundary information of the whole building shape is more important than that of the building shape, then the value of φ needs to be increased [39]. In this research, the boundary information of both the building shape and the whole building shape is very important, and the value of φ is set to 0.5.

5.2.6 Post-Processing

Output obtained from the proposed multi-task Res-U-Net model with attention mechanism is in binary format, in which every pixel indicates the category of the building roof or the background for task 1, and indicates the category of the whole building shape or the background for task 2.

Since raster data for segmentation of the building roofs and the whole building shapes are not very useful for spatial analysis and spatial calculation, an operation of post-processing was applied in this chapter. In post-processing, raster data are taken as input, and spatial boundaries of the building roofs and the whole building shapes are generated as vector format after noise removal and separation of connected objects.

5.2.6.1 Noise Removal

For input images, pixels of the building roof and the whole building shape are represented in white, and pixels of the background are represented in black, which are shown in the segmentation part of Figure 5.4. To remove the noises of detected

building roofs and building shapes, we apply common morphological operations on the binary images. We use the operation of erosion first, then the operation of dilation, which can remove small noises automatically. Then we set a threshold about area, and small noises are removed based on the threshold.

5.2.6.2 Distance Transform

To separate the connected building roofs or building shapes, we assume that the area of the connected part is smaller than the area of the building roof or the whole building shape itself. Based on this fact, distance transformation [110] is applied to the binary images. Therefore, the connected part is given a smaller weight in comparison with the building roof or the whole building shape itself.

5.2.6.3 Local Maxima

After the operation of distance transformation, local maxima method [110] is introduced. Since the connected part is smaller than the building roof or the whole building shape itself, finding local maxima can make sure that it is within the building roof or the whole building shape, and not within the connected part. And the local maximum point is used as the input sink of the watershed segmentation.

5.2.6.4 Watershed Segmentation

Watershed segmentation is widely used in image segmentation. This method treats the image like a topographic map, with the brightness of each point representing its height, and finds the lines that run along the tops of ridges [162]. Local maxima are used as the sink points, and the negative of the distance transform is regarded

as the cost map. This method can be used to separate the connected binary blobs with high efficiency.

5.2.6.5 Vectorization and Smoothing

Raster images obtained from watershed segmentation are vectorized based on libraries of OpenCV2[4] and GDAL/OGR[5]. In the raster images, every pixel contains the geographical information, and raster images are converted into vector files with correct overlaying (Shapely[6] library). When converting data from raster into vector, results have redundant vertices and noises. Algorithm of Douglas-Peucker [163] is used to simplify vector files. This algorithm makes it possible to maintain the general shape of the geometry when redundant vertices are simplified. Moreover, basic properties (e.g., area and perimeter of spatial boundaries) are automatically recorded.

5.2.6.6 Polygon Matching

After getting polygons of the building roofs and the whole building shapes, the process of polygon matching can help us detect each pair of the building roof and the whole building shape. In order to solve this problem, Rtree data structure is introduced. The main idea of the Rtree data structure is to group nearby objects and represent them with their minimum bounding rectangle in the next higher level of the tree. This data structure can effectively accelerate the nearest neighbor search for spatial data. In this research, for every polygon of the building roof, it will recall all polygons of the building shapes that intersect it based on the Rtree data

[4]https://opencv.org/opencv-2-4-8/

[5]https://gdal.org/

[6]https://shapely.readthedocs.io/en/stable/manual.html

structure. Among all candidate polygons of the building shapes, only the one that has the largest overlap area with the polygon of the building roof is recognized as the corresponding polygon of the building shape.

5.2.7 Offset Vector Method

Based on the post-processing in Section 5.2.6, raster images can be converted to vector polygons. For every remote sensing image, two sets of vector polygons can be generated, i.e., a set of vector polygons of the building roofs $\Psi_r = \{poly_1^r, poly_2^r, \cdots, poly_n^r\}$, and a set of vector polygons of the whole building shapes $\Psi_s = \{poly_1^s, poly_2^s, \cdots, poly_n^s\}$. Then based on the offset vector method, a set of vector polygons of the building footprints $\Psi_f = \{poly_1^f, poly_2^f, \cdots, poly_n^f\}$ can be calculated.

As noted in Section 5.1, in most instances, the building footprint has the same shape as the building roof, except for a certain offset in position. After getting the vector polygons of the building roof and the whole building shape, the offset vector helps us to move the polygon of the building roof to the position of the building footprint. Suppose $poly_i^r$ is the ith polygon of Ψ_r, $poly_i^s$ is the corresponding ith polygon of Ψ_s, $poly_i^f$ is set to the corresponding ith polygon of Ψ_f, and $\overrightarrow{v_i}$ is the offset vector from $poly_i^r$ to $poly_i^f$. Then C_i^r represents the centroid of $poly_i^r$, and C_i^s represents the centroid of $poly_i^s$. Both C_i^r and C_i^s lie on the offset vector $\overrightarrow{v_i}$, therefore, $\overrightarrow{v_i}$ can be calculated by C_i^r and C_i^s. When the polygon of $poly_i^r$ moves along the vector of $\overrightarrow{v_i}$, the critical position is defined as $poly_i^r$ is about to move out $poly_i^s$, and the position of $poly_i^r$ is exactly the polygon of $poly_i^f$.

To understand this method easier, a flowchart is shown in Figure 5.5. The green polygon is denoted by $poly_r$, which represents the polygon of the building roof. The blue polygon is denoted by $poly_s$, which represents the polygon of the whole building shape. Then $poly_r$ and $poly_s$ are overlaid together with the same coordinate system.

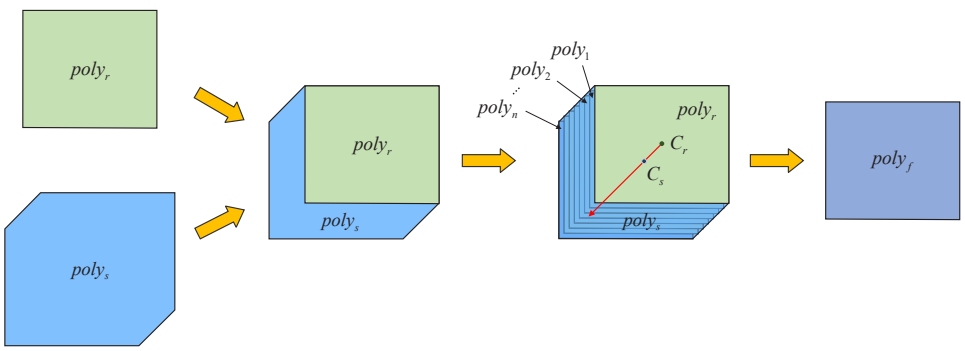


Figure 5.5: Working principle diagram of the offset vector method.

The third step shows how the offset vector method works. C_r denotes the centroid of $poly_r$, and C_s denotes the centroid of $poly_s$. Based on C_r and C_s, the offset vector is created, which is shown as the red vector in the figure. Then $poly_r$ is moved along the offset vector with the same incremental step size every time, and a set of polygons are generated, i.e., $\Phi_r = \{poly_1, poly_2, \cdots, poly_n\}$. Then the polygon is about to move out $poly_s$ is defined as the polygon of the building footprint, which is denoted as $poly_f$. Finally, $poly_f$ is calculated by the offset vector method based on $poly_r$ and $poly_s$.

The offset vector method is also implemented by Algorithm 5. For the input of this algorithm, $poly_r$ denotes the polygon of the building roof, and $poly_s$ is the polygon of the corresponding building shape. The parameter of *step* represents the incremental step size, and *threshold* refers to a value to determine whether the polygon is about to move out $poly_s$ or not. These two parameters can be set to different values for different situations. For the offset vector method, on Lines 2−5, we retrieve the centroid of $poly_r$, i.e., (x_r, y_r), and the centroid of $poly_s$, i.e., (x_s, y_s). On Line 6 and Line 7, the offset vector is calculated based on the two centroids, which is denoted as $(vector_x, vector_y)$. When $poly_r$ is moved along the offset vector, a series of polygons will be generated, and $poly_i$ denotes the ith polygon of them. For $poly_i$, $area\_inter_i$ refers to the area of intersection between $poly_i$ and $poly_s$, and $area\_inter$ means the area of intersection between $poly_r$ and $poly_s$. Then, on

Lines 12−17, a while loop is used to find the polygon of the building footprint. If the value of $area\_inter_i/area\_inter$ is greater than or equal to $threshold$, the while loop continues, otherwise, the while loop stops. In each iteration, $poly_i$ moves the distance of one step size ($step$) along the offset vector on Lines 13−16. The while loop stops when the value of $area\_inter_i/area\_inter$ is less than $threshold$. Finally, the last $ploy_i$ is set to $poly_f$ on Line 18, and $poly_f$ is returned as the polygon of the building footprint. The detailed proof procedure of the offset vector method is presented in the appendix section.

Based on the offset vector method, polygons of the building footprints can be calculated from the corresponding polygons of the building roofs and the whole building shapes. For extensive studies of building footprint extraction [35–37,164], the building roofs are recognized as the building footprints, while in this research, the offset vector method is used to detect building footprints based on building roofs and the whole building shapes. For the generalization of the method, different types of remote sensing data can provide more data sources, and high-resolution remote sensing images tend to provide higher semantic segmentation accuracy. This method can be applied in different types of remote sensing images. This method can also be applied in other places or countries with high-rise buildings, especially, the residential complexes. For buildings within residential complexes, their shapes are more regular, and our method can detect these building footprints properly. For some special scenarios, the limitation of this method is that it cannot be applied to those buildings where the shapes of the building roofs and the building footprints are very different, such as buildings with pointed roofs and special form buildings.

Algorithm 5: Offset Vector Method

Input: $poly_r, poly_s, step, threshold$

Output: $poly_f$

begin

$x_r \leftarrow poly_r.centroid.x$

$y_r \leftarrow poly_r.centroid.y$

$x_s \leftarrow poly_s.centroid.x$

$y_s \leftarrow poly_s.centroid.y$

$vector_x \leftarrow x_s - x_r$

$vector_y \leftarrow y_s - y_r$

$area\_inter \leftarrow area(poly_r \cap poly_s)$

$i \leftarrow 0$

$poly_i \leftarrow poly_r$

$area\_inter_i \leftarrow area(poly_i \cap poly_s)$

while $area\_inter_i / area\_inter \geq threshold$ **do**

$i \leftarrow i + 1$

$offset_x \leftarrow vector_x \times step \times i$

$offset_y \leftarrow vector_y \times step \times i$

$poly_i \leftarrow translate(poly_r, offset_x, offset_y)$

$area\_inter_i \leftarrow area(poly_i \cap poly_s)$

$poly_f \leftarrow poly_i$

return $poly_f$

5.2.8 Name Parsing of POIs

Based on the process of Section 5.2.7, the spatial boundaries of POIs are detected from the building roofs and the whole building shapes, and the next step is to parse names of POIs. Historical order dataset from Meituan platform consists of rich

geospatial information about customers and riders. Suppose $poly_i^f$ is one polygon of the building footprint from Ψ_f in Section 5.2.7, and $O_i = \{o_1, o_2, \cdots, o_m\}$ is a set of orders with customers' geographic coordinates located in the boundary of $poly_i^f$. Based on the natural language processing technology, names of POIs can be parsed from delivery addresses of O_i. Then $N_i = \{n_1, n_2, \cdots, n_m\}$ denotes the set of parsed names of POIs from O_i. GPS drift refers to the difference of the real location and the location recorded by a GPS receiver, and it makes some GPS points locate in the wrong locations. Therefore, multiple POI names (i.e., N_i) can be found in $poly_i^f$, but only the most frequent name in N_i is recognized as the name of $poly_i^f$. Based on this method, POI names can be easily parsed from the historical order dataset.

5.3 Numerical Experiments

In this section, we first outline the dataset of the study, followed by the data augmentation and experimental settings. Evaluation metrics and experimental results of the proposed model are described in the end.

5.3.1 Data Descriptions

For existing publicly available remote sensing building datasets [165–167], they focus on low-rise buildings and the building outline can be generally recognized as the building footprint. But in our study, we focus on high-rise buildings, and the building roofs and the whole building shapes should be extracted separately. Therefore, we did a lot of work on labelling the data for training and testing. The data were obtained from remote sensing images of urban areas of Shenzhen, China in Google Earth. These remote sensing images include three bands, and the format is geotiff with geographic information. In this chapter, we collected 108 remote sens-

Table 5.1: The dataset division.

	Numbers of Samples	Numbers of buildings
Training dataset	256	8,553
Validation dataset	88	2,891
Test dataset	88	3,067

ing images sized 1,024×1,024 pixels. For every image, we manually outlined all the building roofs and the whole building shapes in it, respectively, which is illustrated in Figure 5.6. In this chapter, remote sensing images were labelled by the VGG Image Annotator (VIA) tool [168]. For building roofs, most of them have relatively simpler boundaries, such as quadrangle. While for building shapes, most of them have relatively complex boundaries, and one of their edges have the same shapes as the corresponding edges of the building roofs. During the labelling process of the building shapes, we first copied the corresponding boundaries of the building roofs, then moved them to the right places and started to outline the boundaries of the building shapes. Therefore, we created a more accurate ground truth dataset based on these characteristics. For both the building roofs and the whole building shapes, annotated images were first saved as the JavaScript Object Notation (JSON) format, and then were transferred into binary images with only black and white colours. Finally, every remote sensing image was split into 4 images with the size of 512×512, and the same operation was also performed on the binary images. The dataset is available at https://github.com/BuildingFootprint/POI-Detection-of-High-Rise-Buildings-Using-Remote-Sensing-Images-Dataset.

In the experiment, the whole dataset covers 14,511 building objects, polygons of the building roofs and the whole building shapes are seen as the ground truth for training and evaluating models. The dataset is split into three sets: the training dataset, the validation dataset, and the test dataset with ratios of 60%, 20%, and 20%. The dataset division is illustrated in Table 5.1.

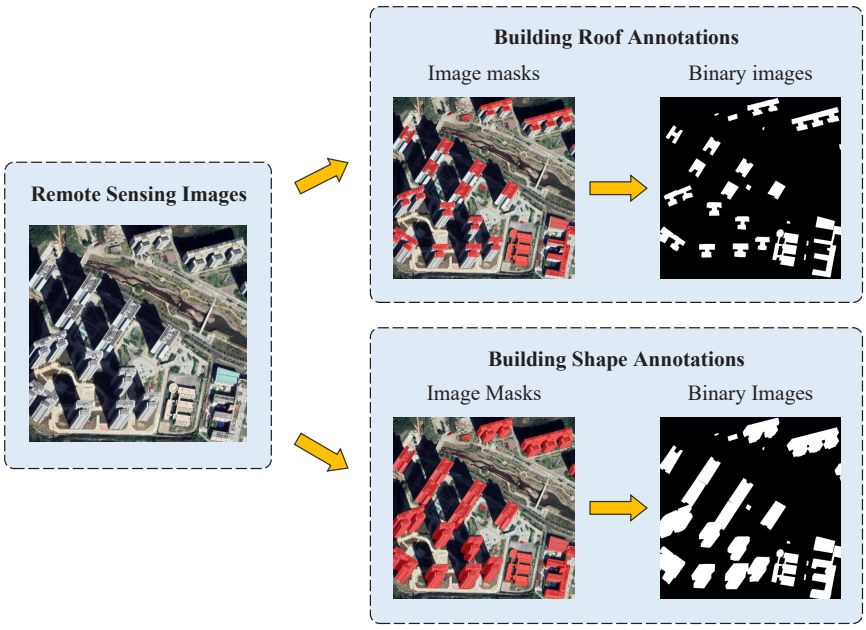


Figure 5.6: Annotation diagram of remote sensing images.

5.3.2 Data Augmentation

Deep CNNs need a large number of training data, which may be unavailable at the training stage. Data augmentation of images is critical to make the network invariant and robust, and to avoid overfitting if the training samples are not large enough. If data augmentation is applied, the model always sees the different sets of images during every iteration. In this chapter, various methods of data augmentation are applied as follows. For every pair of input images and ground truth binary images, they were flipped horizontally or vertically randomly based on a probability of 0.5. Images were offset to left, right, up, or down with a random number of pixels. Moreover, images were rotated with a random angle and a random center, and zoomed in or out based on a random scale factor. Finishing the transformation, the rest pixel values of images were set to 0s. Then these augmented images were fed into models as original training images.

5.3.3 Experimental Settings

The multi-task Res-U-Net model with attention mechanism was built by Keras based on the Tensorflow backend. In this chapter, the size of input images for every experiment was $512 \times 512 \times 3$. The size of the output images for every experiment was $512 \times 512 \times 1$, and outputs were binary images. During the training and testing processes, all images were converted to tensors for computation. Our proposed model and other baseline models were all trained on Nvidia Tesla M60 (4 GPUs with 16GB memory each).

All hyper-parameters used in these experiments were optimal parameters in comparison with repeated test results. During the training stage, the Adam algorithm was chosen as the optimizer, and the initial learning rate was set with 10^{-4}. Weights and parameters of models were initialized by the Glorot normal initializer [169]. For all models, the epoch was set to 50, and the mini-batch size was set to 4.

5.3.4 Evaluation Metrics

To quantify the performance of models, precision, recall, F1-score, and IoU are utilized as qualitative metrics. In semantic segmentation, precision represents the proportion of correctly classified positive pixels in all pixels predicted as positive. Recall is defined as the proportion of correctly classified positive pixels in all true positive pixels. F1-score is the harmonic mean of precision and recall. IoU is used to evaluate the accuracy based on the segment level. Definition of the four metrics is represented as follows:

$$\text{Precision} = \frac{N_{TP}}{N_{TP} + N_{FP}}, \tag{5.9}$$

$$\text{Recall} = \frac{N_{TP}}{N_{TP} + N_{FN}}, \tag{5.10}$$

$$\begin{aligned} \text{F1-score} &= \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \\ &= \frac{2 \cdot N_{TP}}{2 \cdot N_{TP} + N_{FP} + N_{FN}}, \end{aligned} \tag{5.11}$$

$$\text{IoU} = \frac{N_{TP}}{N_{FP} + N_{TP} + N_{FN}}, \tag{5.12}$$

where N_{TP} is the number of overlapping building roof (or building shape) pixels in predicted and ground-truth images (true positive). N_{FP} is the number of building roof (or building shape) pixels in the predicted image, but not in the ground-truth image (false positive). N_{FN} is the number of building roof (or building shape) pixels in the ground-truth image, but not in the predicted image (false negative).

5.3.5 Experimental Results

In this chapter, we divided experiments into three stages. The first stage was to extract polygons of the building roofs and the whole building shapes based on the multi-task Res-U-Net model with attention mechanism. The second stage was to post-process polygons of the building roofs and the whole building shapes, and detect polygons of the building footprints based on the offset vector method. The third stage was to combine POI information with the building footprints based on historical OFD orders.

Detailed results for the proposed model are given in Table 5.2. For every evaluation metric, the performance of the whole building shape segmentation is better than

Table 5.2: Evaluation results of the proposed model.

Index	Building Roof	Building Shape	Overall
Precision	75.45%	77.35%	76.40%
Recall	79.92%	82.05%	80.99%
F1-score	77.05%	79.02%	78.04%
IoU	63.55%	66.05%	64.80%

that of the building roof segmentation. F1-score and IoU are two important indices for evaluating the performance of semantic segmentation. For the building roof segmentation, F1-score and IoU are 77.05% and 63.55%, respectively. For the whole building shape segmentation, F1-score and IoU are 79.02% and 66.05%, respectively. Overall, F1-score and IoU are 78.04% and 64.80%, respectively.

Figure 5.7 illustrates a sample diagram of the segmentation results based on the test dataset. We take three cases (i.e., Case 1, Case 2, and Case 3) as examples to illustrate the results. For every case, we use two rows to show results of the building roof segmentation and the whole building shape segmentation, respectively. In order to show the results clearly, we use red masks to represent the predicted building roofs, and green masks to represent the predicted whole building shapes.

Compare the predicted segmentation results with the ground truth, we can still find some problems. In Case 1, the narrow gaps between closely neighbored buildings are annotated in the ground truth, but our proposed model is incapable of expressing such fine details and separate these kinds of buildings, e.g., segmentation results in the blue circles. Within the red circles, a part of the road is wrongly recognized as a building, and within the magenta circles, buildings under construction are detected, but they are not included in the ground truth. In Case 2, compared with the ground truth, polygons of predicted results always have curve boundaries, which are shown in the orange ellipses. In the green ellipses of Case 3, we can see that shadows

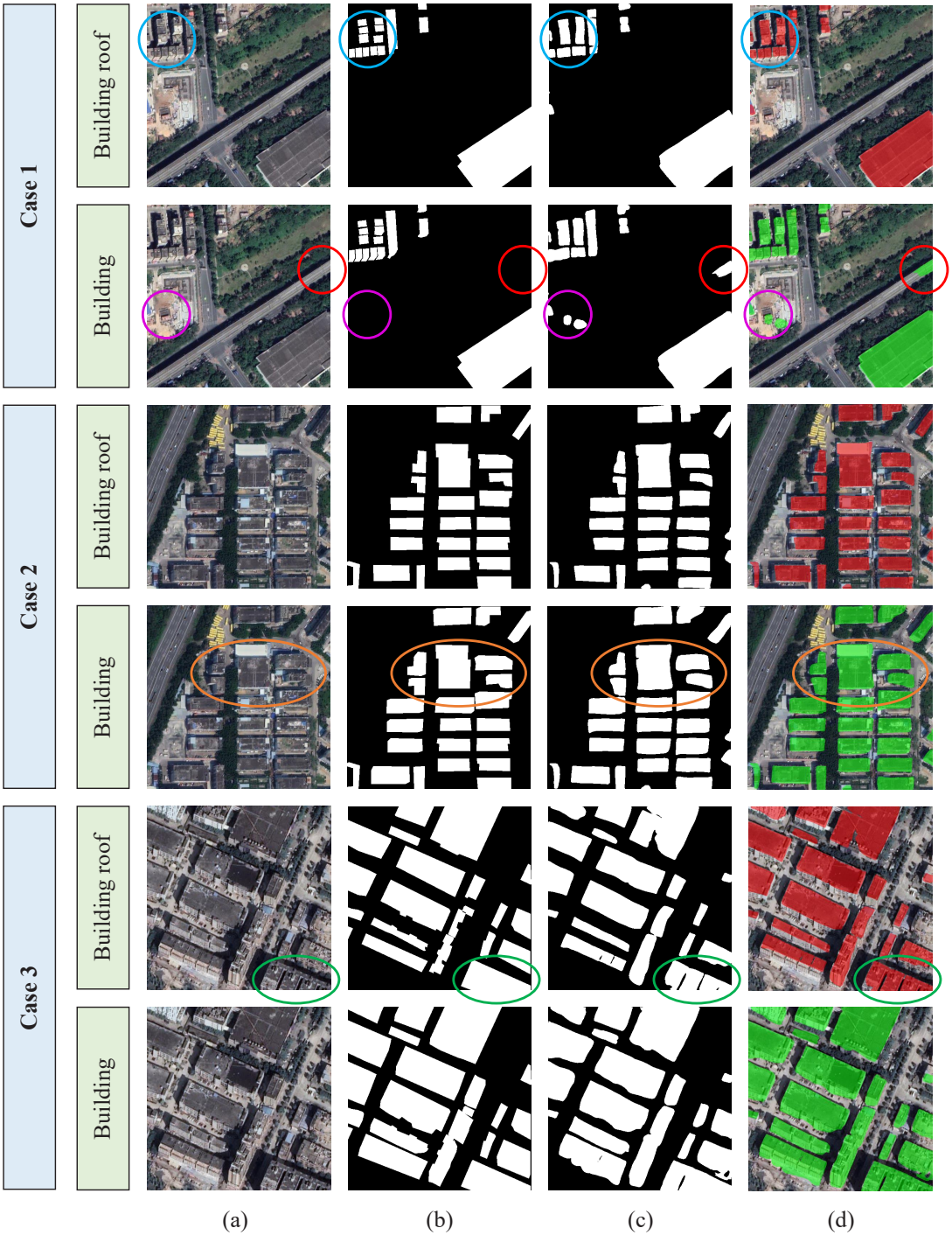


Figure 5.7: Results of the building roof extraction and the whole building shape extraction on the test dataset. (a) Original remote sensing images. (b) Ground truth images of the building roof and the whole building shape. (c) The predicted binary images of the building roof and the whole building shape based on our model. (d) Overlapping display of the predicted masks with the original remote sensing images.

of buildings are excluded in the segmentation results, but they are included in the ground truth.

After detecting polygons of the building roofs and the whole building shapes, we need to match every building roof polygon to the corresponding building shape polygon. To increase the matching efficiency, we introduced R-tree data structure, which is utilized for the effective storage of spatial data indices. Then based on the offset vector method mentioned in Section 5.2.7, polygons of the building footprints can be calculated from polygons of the building roofs and the whole building shapes.

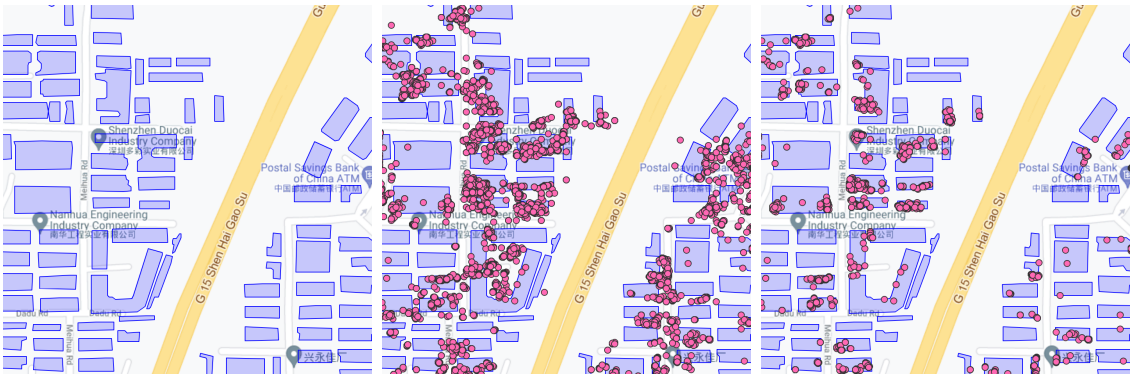
An example is illustrated in Figure 5.8 to explain this process. In Figure 5.8(a), the red line represents the spatial boundary of the building roof and the background is the remote sensing image. The green line in Figure 5.8(b) represents the spatial boundary of the building footprint. In Figure 5.8(c), spatial boundaries of the building roofs and footprints are overlaid together, and most of the building roofs have an offset distance from the corresponding building footprints. However some spatial boundaries of building roofs and building footprints overlap with each other, which means the offset distance of these cases is 0m. After calculation, in this research area, the largest offset distance is 9.08m, and the average offset distance is 3.70m. The value of the offset distance varies with the different view angles of remote sensing sensors and different heights of buildings.

Then name parsing of POIs is processed based on OFD data, and the whole procedure is shown in Figure 5.9. Figure 5.9(a) illustrates building footprints in the research area, and every polygon represents a building footprint. In Figure 5.9(b), customers' GPS locations of historical OFD order data are overlaid on the map, and every pink point represents the customer's GPS location of an order. All GPS locations located in the research area are shown in this figure. Then the next step, only orders with GPS locations located in polygons of the building footprints are kept, which is shown in Figure 5.9(c), and other orders are removed. Finally, the

(a) (b) (c)

Figure 5.8: Spatial boundaries of the building roofs and footprints. (a) Spatial boundaries of the building roofs. (b) Spatial boundaries of the building footprints. (c) Spatial boundaries of the building roofs and footprints.

most frequent POI name of orders located in every building footprint is recognized as the name of the building footprint. As shown in the figure, an issue is that some building footprints cannot recall any orders, and an effective method to solve this problem is to collect OFD order data for a long period of time.



(a) (b) (c)

Figure 5.9: Name parsing of POIs. (a) Spatial boundaries of the building footprints. (b) Spatial boundaries of the building footprints and GPS points located in the research area. (c) Spatial boundaries of the building footprints and GPS points located in them.

5.4 Discussion

5.4.1 Comparison with Different Methods

For the comparison with the performance of the proposed model, we use six mainstream remote sensing image segmentation models, including FCN-8s [170], U-Net [97], Res-U-Net [112], SegNet [103], DeepLabV3+ [100], and DeConvNet [101], as these networks are very representative and effective for semantic segmentation based on remote sensing images.

As mentioned in Section 5.3.4, precision, recall, F1-score, and IoU are metrics that can illustrate the performance of semantic image segmentation. Quantitative results of four metrics are provided in Table 5.3 and 5.4. The bold values of F1-score and IoU represent the best results among these models. Overall, the proposed model outperforms other deep CNNs in both the building roof segmentation and the whole building shape segmentation. For the proposed model, the F1-score and IoU of the building roof segmentation are 77.05% and 63.55%, respectively, which are the highest among all models; these two values of the whole building shape segmentation are 79.02% and 66.05%, respectively, which are also the highest among all models. Moreover, the classification accuracy of the whole building shape is better than that of the building roof. Res-U-Net achieves the best precision (76.96%) in the building roof segmentation, which is higher than that of our proposed model (75.45%). Compared with our proposed model, SegNet achieves relatively high performance of precision in both the building roof segmentation and the whole building shape segmentation, which are 76.53% and 77.61%, respectively. For both the building roof segmentation and the whole building shape segmentation, the performance of F1-score and IoU of SegNet is the highest among all the baseline models, with F1-score and IoU of 75.70% and 63.24% for the building roof segmentation and with F1-score

Table 5.3: Quantitative comparison of semantic segmentation for the building roof based on FCN-8s, U-Net, Res-U-Net, SegNet, DeepLabV3+, DeConvNet, and the proposed model in terms of Precision, Recall, F1-score, and IoU, where the bold values represent the best for corresponding metrics.

Model	Building Roof			
	Precision	Recall	F1-score	IoU
FCN-8s	70.28%	77.85%	73.19%	58.72%
U-Net	74.60%	79.35%	75.54%	62.15%
Res-U-Net	75.48%	79.76%	76.34%	63.48%
SegNet	77.61%	76.51%	76.49%	64.11%
DeepLabV3+	75.21%	62.75%	67.41%	51.48%
DeConvNet	76.24%	67.26%	72.02%	56.19%
Our Model	77.35%	82.05%	**79.02**%	**66.05**%

and IoU of 76.49% and 64.11% for the whole building shape segmentation. Compared with other baseline models, DeepLabV3+, and DeConvNet have relatively worse performance of F1-score and IoU for both the building roof segmentation and the whole building shape segmentation.

To sum up, the proposed model has been compared with different baseline models, and proved to be more effective. As illustrated in Table 5.3 and 5.4, compared with baseline models, the performance of the proposed model is the best.

5.4.2 Ablation Study

Apart from the comparison results of different models, we also focus on the efficiency of every component of our proposed model. Therefore, the ablation study of our proposed model is conducted to explore how these components impact the performance of the semantic segmentation. To make sure an equitable experimental comparison, the ablation study is conducted in the exact same environment as

Table 5.4: Quantitative comparison of semantic segmentation for the whole building shape based on FCN-8s, U-Net, Res-U-Net, SegNet, DeepLabV3+, DeConvNet, and the proposed model in terms of Precision, Recall, F1-score, and IoU, where the bold values represent the best for corresponding metrics.

Model	Building Shape			
	Precision	Recall	F1-score	IoU
FCN-8s	70.68%	77.31%	72.92%	58.69%
U-Net	72.50%	75.84%	73.22%	59.71%
Res-U-Net	76.96%	73.71%	74.13%	60.74%
SegNet	76.53%	76.21%	75.70%	63.24%
DeepLabV3+	68.40%	69.57%	68.04%	52.78%
DeConvNet	72.95%	62.29%	65.89%	50.44%
Our Model	75.45%	79.92%	**77.05**%	**63.55**%

Table 5.5: Quantitative comparison of ablation study for the building roof in terms of Precision, Recall, F1-score, and IoU, where the bold values represent the best for corresponding metrics.

Model	Building Roof			
	Precision	Recall	F1-score	IoU
Model-A	75.35%	77.29%	74.33%	61.05%
Model-R	75.34%	75.24%	74.26%	60.02%
Model-M	75.35%	76.94%	74.87%	61.04%
Our Model	75.45%	79.92%	**77.05**%	**63.55**%

our major experiments mentioned in Section 5.3.3. We perform the ablation study on three variations of our proposed model: model without the attention module, model without the ResNet block, and model without the multi-task strategy. The results of the ablation study are summarized in Table 5.5 and 5.6, where Model-A represents the model without the attention module, i.e., model of multi-task Res-U-Net; Model-R represents the model without the ResNet block, i.e., model of

Table 5.6: Quantitative comparison of ablation study for the whole building shape in terms of Precision, Recall, F1-score, and IoU, where the bold values represent the best for corresponding metrics.

Model	Building Shape			
	Precision	Recall	F1-score	IoU
Model-A	77.94%	79.43%	76.76%	64.16%
Model-R	76.51%	77.67%	76.05%	62.47%
Model-M	75.41%	81.03%	76.95%	64.30%
Our Model	77.35%	82.05%	**79.02**%	**66.05**%

multi-task U-Net with attention mechanism; Model-M represents the model without the multi-task strategy, i.e., model of Res-U-Net with attention mechanism. From the table, one can observe that the proposed model achieves the best performance for both the building roof segmentation and the whole building shape segmentation compared with Model-A, Model-R and Model-M. Therefore, the attention module, ResNet block and the multi-task strategy all play significant roles in our proposed model, and help improve the performance of the semantic segmentation compared with U-Net.

5.5 Chapter Summary

In this chapter, we propose a multi-task Res-U-Net model with attention mechanism to increase the performance of semantic segmentation of the building roofs and the whole building shapes, and apply a sequence of post-processing methods and the offset vector method to detect building footprints of high-rise buildings from remote sensing images. Based on historical order dataset from OFD platform known as Meituan platform, we also identify POI names of the detected building footprints. After designing the attention module, the ResNet block and the multi-task strategy

in the structure of the network, the capability to enhance the model's performance and optimize boundaries of segmentation results can be achieved. Compared with the existing models, our model achieved the best performance in terms of both F1-score and IoU. The results show that the attention mechanism, the ResNet block and the multi-task strategy could effectively increase the amount of information and improve the predictive capability of the model.

By following the extraction of building footprints for OFD services, another important ground target is the road network. For the next chapter, interior road network within residential complexes is extracted based on a novel decoder fusion model using remote sensing images and GPS trajectories.

Chapter 6

Interior Road Extraction Using Multi-Source Data

[7]This chapter develops a decoder fusion model based on dilated Res-U-Net which fuses the remote sensing images and GPS trajectories in a more efficient way to extract the interior road network. The DF-DRUNet model is built on two components. First, two independent dilated Res-U-Net models with each taking remote sensing images and GPS trajectories as input modalities respectively. Second, we fuse the decoders from two modalities based on a dual fusion module, which can help to learn the selection from these two modalities. Based on the interior road extraction from the DF-DRUNet model, we also develop various refinement strategies, i.e., noise removal, skeleton extraction, topology construction, and vectorization. Numerical experiments are conducted from the real dataset of remote sensing images and GPS trajectories.

[7]Parts of this chapter have been submitted in Li, B., Gao, J., Chen, S., Lim, S., and Jiang, H. (2022). Interior Road Extraction within Residential Complexes: A Decoder Fusion Model Leveraging Remote Sensing Images and GPS Trajectories. *IEEE Transactions on Geoscience and Remote Sensing*.

6.1 Introduction

With the development of the mobile Internet, OFD services are becoming more and more popular in people's day-to-day lives [22]. In China, 90% of the deliveries end in residential complexes, which are composed of 10-30 individual buildings and surrounded by their own security fences. Although urban roads are pretty accurate, interior roads within residential complexes still cannot be provided with complete information by traditional map services because they are not open to the general public. However, interior roads within residential complexes play a very important role in route planning of OFD services.

Fig. 6.1 illustrates an example of an interior road network where the solid black line represents the border of the residential complex. The gray polygons represent building footprints, and thick light blue lines represent the interior road network within the residential complex. If there is no interior road network within a residential complex, the OFD platform can only direct the deliverers to the location of the entrance area (e.g., the north entrance or the south entrance in Fig. 6.1), and then the deliverers have to find the customer's place (e.g., the red point in Fig. 6.1) all by their familiarity with the place, which is inefficient and a waste of time. For example, there are 26 buildings within the residential complex in the figure. If the deliverers can only be navigated to the north entrance or the south entrance, then they have to find the destination all by themselves. From Fig. 6.1, Route 1 is a better choice to the destination, but without the information of the interior road network within the residential complex, a route to the north entrance may be planned by the OFD platform, and the deliverers may find the customer's place by Route 2, which is obviously not very efficient and a waste of time.

For OFD services in China, the average delivery time within the residential complex accounts for 30% of the total delivery time, even though its delivery distance is

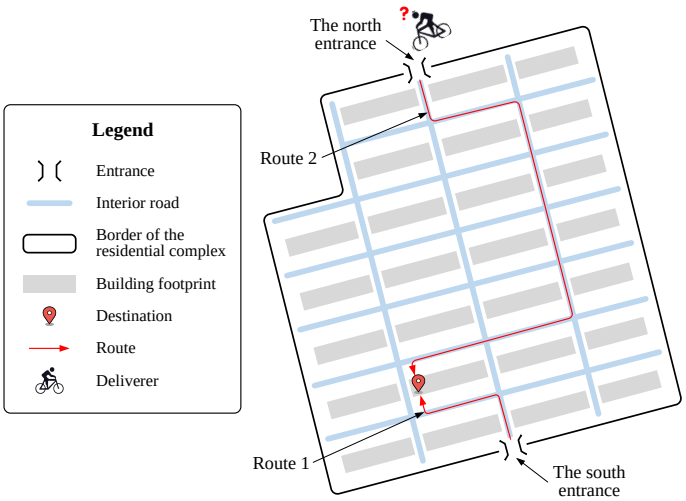


Figure 6.1: An example of interior road network.

only 5% of the total delivery distance. If the road network is complete for both the outside and the inside of the residential complex, then a more efficient route to the customer's destination can be planned by OFD platforms. Therefore, OFD platforms need to fill this research gap of interior road extraction within residential complexes and the accurate interior road information can also improve the delivery efficiency of OFD services.

With the fast growth of GPS technology, more spatio-temporal data can be collected from different kinds of vehicles and these data can be used to extract road networks. For road networks outside the residential complexes, it is easier to acquire because of the availability of large numbers of vehicles. However, for interior road networks within the residential complexes, it is difficult to collect the spatio-temporal data since most of the vehicles are not allowed to enter the residential complexes based on safety considerations. As mentioned above, GPS trajectories generated by deliverers can help us to extract the interior road network within the residential complexes.

In the past few years, a wide range of methods have been developed in the research area of road extraction from GPS trajectories. Although substantial progress has been achieved [38, 39], this research field still faces great challenges. (1) Due to

uneven quality of GPS devices and occlusions by tall buildings and trees, GPS trajectories contain a lot of noises, as illustrated in Fig. 6.2(a). Despite different pre-processing methods have been applied, there still exists the noise problem. (2) Some areas of non-interior roads (e.g., the parking lot in Fig. 6.2(c)) also show some GPS trajectory points and they can be easily mistaken for interior roads without extra information. (3) GPS trajectories could only cover the interior road segments that deliverers have passed, therefore, some interior road areas have no trajectory information if no deliverers have passed, as shown in Fig. 6.2(d).

For existing studies, a wide range of approaches have been developed for road extraction leveraging remote sensing images. Early studies mostly extracted hand-made features such as textures and contours, and used shallow models (e.g., support vector machine) to identify road areas. In recent years, deep learning methods are becoming the mainstream of this research area because of their powerful automatic representation learning ability, and have made remarkable success. However, it still remains a very difficult issue to extract road regions using remote sensing images, particularly in situations as follows. (1) Some interior road areas are totally occluded by tall buildings and trees, as illustrated in Fig. 6.2(b). These interior roads are difficult to be detected only based on the visual information of remote sensing images. (2) Some squares and open spaces have a similar appearance to interior roads, as illustrated in Fig. 6.2(c). Without auxiliary information, it can be difficult to distinguish interior roads from these structures [1].

In summary, because of the aforementioned problems of remote sensing images and GPS trajectories, it is still challenging to extract interior road regions based on a single data source. As remote sensing images and GPS trajectories can provide different types of information, they can complement each other for interior road extraction. Remote sensing image-base approaches and GPS trajectory-based approaches have their own advantages and disadvantages. That is, the fusion of these two comple-

(a) Noises (b) Occlusion

(c) Similar appearance (d) No trajectories

Figure 6.2: (a) Although GPS trajectory data can be used to detect roads, excessive noises are introduced at the same time. (b) Interior roads are usually occluded by tall buildings and trees in remote sensing images. (c) Parking lots and open spaces have similar appearances to the interior roads, hence it is not easy to distinguish interior roads to these structure. (d) Only based on information of GPS trajectories, some interior roads with few GPS trajectories are difficult to identify, as illustrated in the yellow rectangle.

mentary data sources can provide an efficient way to take advantage of information for robust interior road extraction. However, the number of related studies [40, 41] that use the two modalities is very limited. In addition, most of these studies directly fuse the input layer with remote sensing images and GPS trajectory feature maps, which is not an ideal strategy for multi-modal fusion methods. In [135], the authors proposed a decoder fusion model based on a gated fusion module, but due to the

limited ability of refinement, the complementarity between aerial images and trajectories is not fully utilized. Moreover, this approach is based on local information for road extraction, which may fail to identify some severely occluded road areas with few GPS trajectory points, as shown in the yellow rectangles of Fig. 6.2(d). When all the information about the entire remote sensing image and GPS trajectories is taken into account, we can accurately infer whether an area is an interior road region or not. Therefore, the extraction of interior road regions needs to consider the local information and the global information simultaneously [1].

For the existing research, various studies attempted to fuse remote sensing images with other data sources, e.g., street view images [171], OpenStreetMap (OSM) data [172] and Light Detection and Ranging (LiDAR) [173–175], to solve the problem of semantic segmentation of urban scenes, which is related to the task of road extraction. However, some methods just simply fuse the feature maps from two data sources, and some others just calculate the average of predictions from two data sources. We expect that such a concatenation or fusion manner cannot fully explore the complementarities of different data sources and cannot solve the problem of information loss from both two data sources. Among the methods, Sun *et al.* [40] developed a new deconvolution strategy named 1D decoder to address the problem of road extraction. But this approach also simply fuses feature maps of remote sensing images and GPS trajectories which leaves room for further improvement of the fusion strategy.

In this chapter, we propose a decoder fusion model named DF-DRUNet, which fuses remote sensing images and GPS trajectories for extracting interior road network. The DF-DRUNet model is built on two components. First, two independent dilated Res-U-Net models with each taking remote sensing images and GPS trajectories as input modalities respectively. Second, we fuse decoders from two modalities based on a dual fusion module (DFM), which can help to learn the modality selection from

these two modalities. Numerical experiments have been conducted based on the DF-DRUNet model and baseline models from the real dataset of remote sensing images and GPS trajectories. The DF-DRUNet model achieves the best performance of F1-score (85.11%) and IoU (74.26%) among all baseline models.

The main contributions of this chapter can be summarized as follows.

- We propose a decoder fusion model with the DFM unit based on two independent dilated Res-U-Net models for semantic segmentation of the interior road from both remote sensing images and GPS trajectories. The DF-DRUNet model achieves the best performance of F1-score and IoU among all baseline models.
- A novel DFM unit is designed to help to learn the modality selection from both the two modalities of remote sensing images and GPS trajectories
- Most of existing studies extract road network outside residential complexes, while our study concentrates on interior road network extraction within residential complexes from multi-source data.

The remainder of this chapter is organized as follows. Section 6.2 presents the detailed architecture of the DF-DRUNet model and strategies of post-processing. In Section 6.3, we describe the numerical experiments and experimental results in detail. Finally, we discuss the conclusion and possible future expectations in Section 6.4.

6.2 Methodology

In this section, we elaborate the decoder fusion model for interior road extraction using remote sensing images and GPS trajectories. We first describe the model

overview, then the dual fusion module and dilated convolutions, finally, followed by the loss function and post-processing.

6.2.1 Model Overview

In this chapter, our proposed approach follows the process of image-based road extraction, and transform this problem into a binary prediction problem at the pixel level [40, 84, 114]. We select the dilated U-Net with residual block (DRUNet) as the independent model due to the remarkable performance in the field of semantic segmentation. The dilated convolutions can extend the convolution operation's receptive field, and the residual block can address the degradation in deep neural networks [176]. For the input data, we resize the whole remote sensing images into 640×640 pixels and the GPS trajectory features are projected into images and also resized into 640×640 pixels. The detailed data pre-processing is described in Section 6.3.2.

The main structure of our proposed approach is illustrated in Fig. 6.3. It contains two independent dilated Res-U-Net models with each taking remote sensing images and GPS trajectories as input modalities respectively. The remote sensing image decoder and the GPS trajectory decoder store the important information as the auxiliary decoders, respectively. We extract feature maps from two decoders with all scales, i.e., $f_I^{(1)}$, $f_I^{(2)}$, $\cdots$, $f_I^{(5)}$ and $f_T^{(1)}$, $f_T^{(2)}$, $\cdots$, $f_T^{(5)}$, then the corresponding feature maps are fused together by the DFM unit. The fusion feature maps with all scales are represented as $f_F^{(1)}$, $f_F^{(2)}$, $\cdots$, $f_F^{(5)}$, respectively. Finally, the pixel-level prediction is generated based on a linear binary predictor.

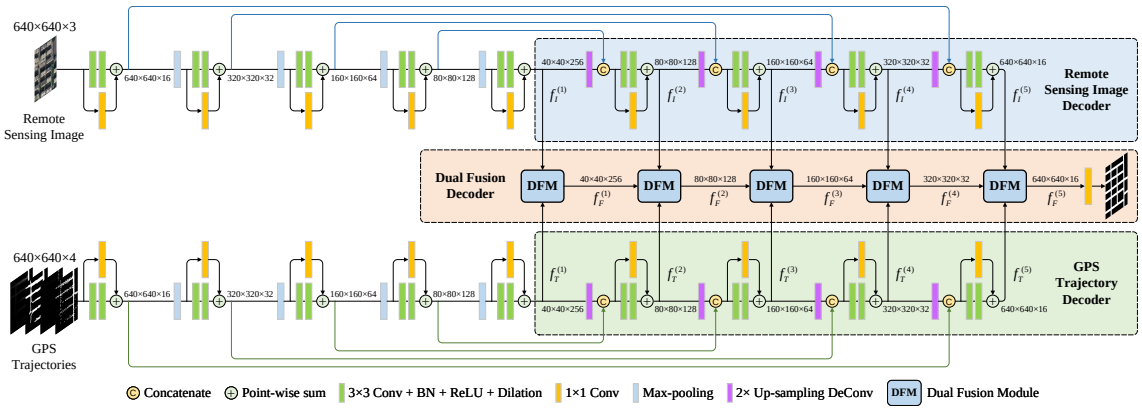


Figure 6.3: The main architecture of the DF-DRUNet model. The model consists of two independent dilated Res-U-Nets. For the input of the remote sensing image, the input dimension is 640×640×3 with three-band colour red, green, and blue. For the input of the GPS trajectories, the input dimension is 640×640×4 with trajectory point density map, trajectory line density map, binary trajectory point map, and binary trajectory line map. The dilated Res-U-Net is based on the U-Net structure, with the repeated application of dilated residual blocks. The dual fusion decoder is fused by the remote sensing image decoder and GPS trajectory decoder through the DFM unit.

6.2.2 Dual Fusion Module

The dual fusion module (DFM) unit is one of the main innovations of the DF-DRUNet model, and it is designed based on the process of human decision making, as shown in Fig. 6.4. A DFM unit consists of four gates, i.e., input gate, selective gate, dual fusion gate, and output gate. Each of these gates can be thought as a neuron in a feed-forward (or multi-layer) neural network. Suppose a task has two independent data inputs, we often choose the more informative one that can provide more valuable input to the task. In other words, for interior road extraction, we prefer to choose data sources that can make the prediction of interior road regions easier and provide more useful information. To be precise, when i is set to 1, the feature map from remote sensing image decoder and the feature map from GPS trajectory decoder are as the inputs. The output is the fused feature $f_F^{(i)} = DFM^{(i)}\left(f_I^{(i)}, f_T^{(i)}\right)$, where $f_I^{(i)}$ is the feature map of remote sensing image decoder

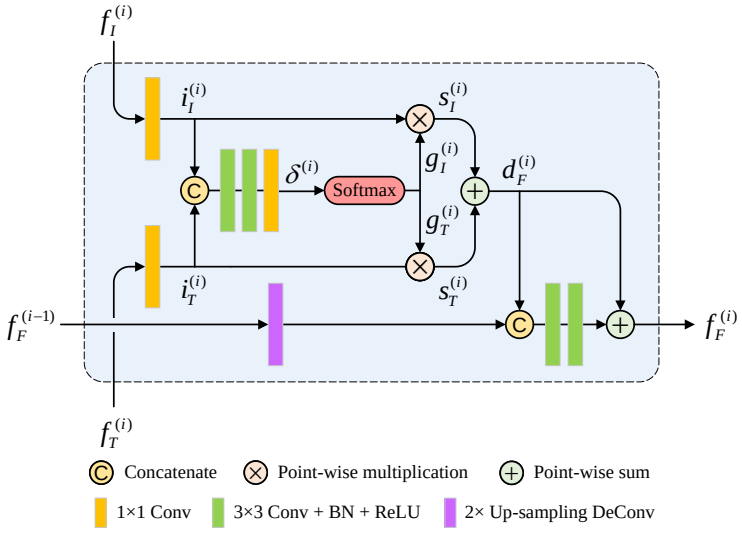


Figure 6.4: The internal structure of a dual fusion module.

and $f_T^{(i)}$ is the feature map of GPS trajectory decoder. When i is set to $2 \sim 5$, it needs to take an extra input, i.e., $f_F^{(i-1)}$ for the previous fusion feature map, and outputs the fused feature $f_F^{(i)} = DFM^{(i)}\left(f_I^{(i)}, f_T^{(i)}, f_F^{(i-1)}\right)$.

Because the two feature maps $f_I^{(i)}, f_T^{(i)}$ are generated from two independent networks, which means they may not have the space consistency, it is not an optimal solution to fuse them directly. Accordingly, the input gate is introduced to transform the input feature spaces $\mathcal{F}_I$ and $\mathcal{F}_T$ to a uniform space $\mathcal{U}$ to make the two feature maps combine linearly in a uniform space. The features are denoted as $i_I^{(i)} = u_I\left(f_I^{(i)}\right) \in \mathbb{R}^{h^{(i)} \times w^{(i)} \times c^{(i)}}$ and $i_T^{(i)} = u_T\left(f_T^{(i)}\right) \in \mathbb{R}^{h^{(i)} \times w^{(i)} \times c^{(i)}}$. Here, $u_I(\cdot)$ and $u_T(\cdot)$ are the channel dimension-preserving operations $\mathbb{R}^{c^{(i)}} \longmapsto \mathbb{R}^{c^{(i)}}$ processed by a 1×1 convolution, which means two different input spaces are linearly transformed into the uniform space $\mathcal{U}$. Input feature maps are transformed into the uniform space, then they are fused properly based on a linear combination, i.e., dual fusion gate, as indicated in Equation (6.1):

$$d_F^{(i)} = g_I^{(i)} \otimes i_I^{(i)} + g_T^{(i)} \otimes i_T^{(i)}, \quad s.t., g_I^{(i)} + g_T^{(i)} = \mathbf{1}, \tag{6.1}$$

where $g_I^{(i)} \in \mathbb{R}^{h^{(i)} \times w^{(i)}}$ and $g_T^{(i)} \in \mathbb{R}^{h^{(i)} \times w^{(i)}}$ denote the selective gate values in terms of the remote sensing image modality and GPS trajectory modality, respectively. $\otimes$ refers to the point-wise multiplication. The constraint of $g_I^{(i)} + g_T^{(i)} = \mathbf{1}$ enforces the DFM unit to generate a fusion feature map with the complementary style. This means that if a region of one modality has a high degree of confidence in prediction, this modality would be given a greater weight, while the other modality would be given a lower weight correspondingly. If both of the two modalities have enough valuable information for the prediction, the weights do not have much effect on the unit itself.

In Fig. 6.4, the selective gate calculates the gate values $g_I^{(i)}$ and $g_T^{(i)}$ based on the information from two input feature maps, i.e., $i_I^{(i)}$ and $i_T^{(i)}$. In order to calculate $g_I^{(i)}$ and $g_T^{(i)}$, firstly the unnormalized prediction $\delta^{(i)} \in \mathbb{R}^{h^{(i)} \times w^{(i)} \times 2}$ is calculated. Then $\delta_I^{(i)} = \delta^{(i)}[:, :, 0]$ and $\delta_T^{(i)} = \delta^{(i)}[:, :, 1]$ are calculated based on $\delta^{(i)}$. Finally, the function of Softmax is used to standardize gate values of pixels in $h^{(i)} \times w^{(i)}$ to satisfy the constraint $g_I^{(i)} + g_T^{(i)} = \mathbf{1}$, as illustrated in Equation (6.2):

$$g_I^{(i)} = \frac{e^{\delta_I^{(i)}}}{e^{\delta_I^{(i)}} + e^{\delta_T^{(i)}}},\ g_T^{(i)} = \frac{e^{\delta_T^{(i)}}}{e^{\delta_I^{(i)}} + e^{\delta_T^{(i)}}}, \tag{6.2}$$

where e denotes the natural logarithm.

The unnormalized gate $\delta^{(i)}$ is calculated from Equation (6.3). Firstly, it is the concatenation of $i_I^{(i)}$ and $i_T^{(i)}$, then implemented by two 3×3 convolution blocks, and finally followed by an 1×1 convolution layer.

$$\delta^{(i)} = \psi_{1\times1}\left(\phi_{3\times3}^2\left(i_I^{(i)} \copyright i_T^{(i)}\right)\right) \in \mathbb{R}^{h^{(i)} \times w^{(i)} \times 2}, \tag{6.3}$$

where $\copyright$ refers to the concatenation operation of $i_I^{(i)}$ and $i_T^{(i)}$, $\phi_{3\times3}^2(\cdot)$ denotes two successive 3×3 convolution layers with Batch Normalization (BN) and Rectified

Linear Unit (ReLU) activation, and $\psi_{1\times1}(\cdot)$ represents a 1×1 convolution layer.

The fused feature maps of the dual fusion gate are $d_F^{(i)}$, which contain more important information from the two input feature maps. Based on the inspiration of the residual refinement learning [177], we use this information directly as the basis, and expect the decoder to learn the residual refinement of $f_F^{(i)}$, as shown in Equation (6.4):

$$f_F^{(i)} = d_F^{(i)} + R\left(d_F^{(i)}, f_F^{(i-1)}\right), \tag{6.4}$$

where $f_F^{(i)}$ refers to the refined features calculated from $d_F^{(i)}$, and $R(\cdot)$ is the residual refinement function.

The residual refinement function is composed by a deconvolutional layer, a concatenation operation, and two 3×3 convolution blocks, that is,

$$R\left(d_F^{(i)}, f_F^{(i-1)}\right) = \phi_{3\times3}^2\left(D_{2\times}\left(f_F^{(i-1)}\right) \copyright\, d_F^{(i)}\right), \tag{6.5}$$

where $D_{2\times}(\cdot)$ denotes the 2× up-sampling operation, and $\phi_{3\times3}^2(\cdot)$ refers to the two successive 3×3 convolutions with batch normalization and followed by a ReLU activation.

6.2.3 Dilated Convolution Module

The dilated convolution is an extension of the standard convolution, in which the convolutional filter is up-sampled by inserting zeros between the weights [178]. Dilated convolution is applied over a two dimensional feature map x, where for every position i and a filter w, the output y is defined as:

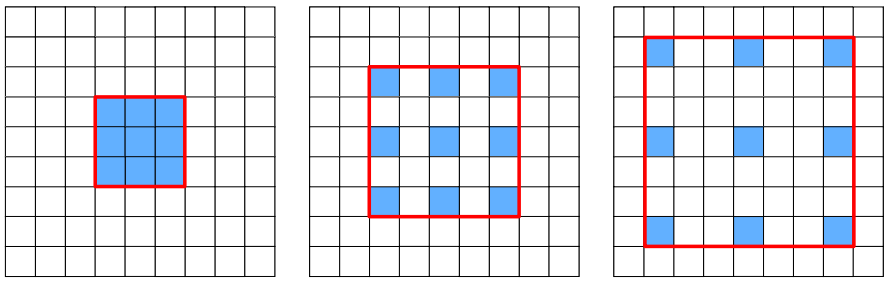


Figure 6.5: Dilated convolutions with different dilation rates. The effective receptive field of a dilated convolution is enlarged by inserting gaps between the kernel weights of a 3×3 filter based on the dilation rate.

$$y[i] = \sum_{k=1} \left(x[i + d \cdot k] \cdot w[k] \right), \tag{6.6}$$

where d denotes the dilation rate, and is similar to the stride of the sampled input signal. This operation is equivalent to convolving the input x with up sampled filters generated by adding $(d-1)$ 0s between the two continuous filter values along every spatial dimension.

The principle of dilated convolution is to expand the filters' view field, which helps capture multiscale information and keeps the spatial resolution. Fig. 6.5 shows dilated convolutions with different dilation rates. When the dilation rate is set to 1, the dilated convolution can be recognized as a typical convolution. If the value of the dilation rate is 2, the dilated convolution has a receptive field of 5×5, which is enlarged from the kernel of 3×3. If the dilation rate is 3, the receptive field is enlarged to 7×7. Therefore, the main advantage of dilated convolution is that it can enlarge the receptive field of the convolution operation without adding other training parameters.

The receptive field describes the area of an image that can be viewed by an artificial neuron to extract information. A large receptive field is needed to learn multiscale

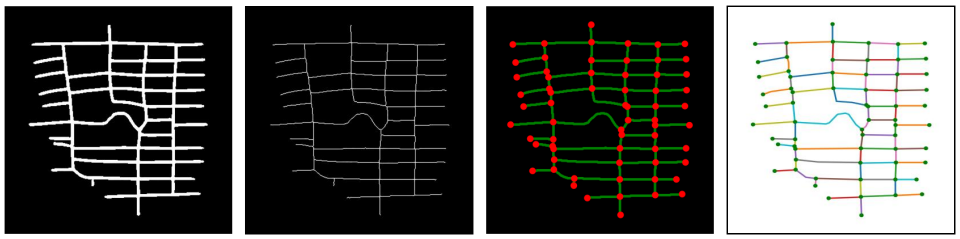
features which is conventionally achieved by connecting successive convolutional layers in a cascade and using max pooling layers to spatially down sample the image [178]. Dilated convolutions allow the CNN to more efficiently learn multiscale features without a re-scaled image and loss of resolution. Cascaded dilated convolutions also expand the receptive field exponentially, whereas, cascaded standard convolutions expands it linearly [179]. Hence, the dilated convolution operations can reduce the semantic gap between encoder's features and decoder's features, and capture the multiscale information.

6.2.4 Post-Processing

Output binary images are generated based on the DF-DRUNet model, and each of these pixels represents a category of the interior road region or the background. Since binary images for segmentation of the interior road regions can not be directly used in spatial analysis and calculation, various strategies of post-processing is adopted. For post-processing, predicted binary images are taken as input, and interior road networks are generated after noise removal, skeleton extraction, topology construction, and vectorization, as shown in Fig. 6.6.

6.2.4.1 Noise Removal

For the input binary images, the interior road region is coloured in white, and the background is coloured in black, as illustrated in Fig. 6.6(a). Since there are usually some small separated noises in the predicted binary images, we introduce the morphological operation for noise removal in this chapter. Continuous operations of erosion and dilation can help to remove these small noises automatically.



(a) (b) (c) (d)

Figure 6.6: Post-processing of predicted interior road regions. (a) Noise removal of the predicted interior road regions. (b) Skeleton extraction of interior road regions. (c) Topology construction of interior road network. (d) Vectorization and smoothing of interior road network.

6.2.4.2 Skeleton Extraction

The binary images generated by Section 6.2.4.1 contain the interior road regions, but the road regions are not suitable for spatial analysis and spatial calculation. Therefore, we have to find the centerlines of the road regions. Skeleton extraction is one of the morphological operations on binary images, and it can reduce the interior road regions to 1 pixel wide representation. In [180], the author introduced a fast parallel algorithm for thinning digital patterns. This method can be used for skeleton extraction and the results are shown in Fig. 6.6(b).

6.2.4.3 Topology Construction

After the interior road centerlines are generated by Section 6.2.4.2, the topological connectivity still can not be guaranteed. The purpose of topology construction is to find all road nodes and road links from the interior road centerlines. In the road network, road links are separated by the road nodes. In [181], the authors presented a python library called Skan for skeleton analysis including topology construction. This operation is implemented by this library and the results are shown in Fig. 6.6(c).

6.2.4.4 Vectorization and Smoothing

In Section 6.2.4.3, we acquire all road nodes and road links of the interior road centerlines from the binary images, and in this section, we would vectorize them and project them into GPS coordinates. For these images, the GPS coordinate of each pixel can be calculated based on the projection information of pre-processing stage, then pixel information of road nodes and road links are converted into GPS coordinates and saved as vector files by the library of Shapely [8]. When binary images are vectorized into interior road networks, some road links have redundant and unsmooth parts. In [163], the authors introduced an algorithm called Douglas-Peucker to help smooth vector links. This algorithm can keep the main shape of road links while smoothing the redundant parts. Fig. 6.6(d) illustrates the interior road network based on the operation of vectorization and smoothing.

6.3 Numerical Experiments

6.3.1 Data Descriptions

In this chapter, our experiments focus on the interior roads within the residential complexes, which are collected in Beijing, China. Particularly, this dataset includes about 80 million GPS trajectory records and remote sensing images of 2,600 different residential complexes. Since the area of each residential complex is different, the size of its corresponding remote sensing image is also different. After calculation, the average height and width of these remote sensing images are both around 1,280 pixels, therefore, all remote sensing images are processed with the size of 1,280×1,280. For each remote sensing image, based on the data pre-processing in Section 6.3.2, we generate four feature maps, i.e., a 1,280×1,280 trajectory point density map, a

Table 6.1: The dataset division.

	Number of samples	Number of trajectories
Training dataset	1,820	5.63×10^7
Validation dataset	390	1.25×10^7
Test dataset	390	1.19×10^7

1,280×1,280 trajectory line density map, a 1,280×1,280 binary trajectory point map, and a 1,280×1,280 binary trajectory line map. The ground truth binary images are manually outlined based on the remote sensing images. Finally, the dataset is split into three datasets: 70% samples are used as the training dataset, 15% samples are used as the validation dataset, and the rest 15% samples are used as the test dataset, as shown in Table 6.1.

6.3.2 Data Pre-processing

In this section, we describe how the multi-source input data are prepared. The remote sensing images can be fed into the proposed model directly, while GPS trajectories have to be converted into feature maps with the same spatial resolution of the remote sensing images. In order to capture more potential information from GPS trajectories, four types of feature maps are generated, which are trajectory point density maps, trajectory line density maps, binary trajectory point maps, and binary trajectory line maps based on raw GPS trajectory data.

Raw GPS Trajectory Sample: With the fast development of OFD services, deliverers' GPS trajectory data can be easily collected and used to build a large scale GPS trajectory database within residential complexes. In this database, every trajectory sample contains important spatio-temporal information, i.e., the longitude, the latitude, and the timestamp of every GPS trajectory point.

Trajectory Point Density Map Generation: For raw GPS trajectory data, the discrete GPS points can not be fed into the deep neural networks directly, therefore, the raw GPS data need to be converted into the 2D trajectory feature maps. The whole generation of GPS trajectory feature maps is illustrated in Fig. 6.7. Specifically, given a set of longitude and latitude information of a residential complex's boundary, we first calculate the bounding box of the boundary, and then search and crop out the remote sensing image within the same geographic area from the remote sensing image database. For the GPS trajectory database, we need to find all GPS trajectory points within the spatial range $[lon_l, lat_l] * [lon_u, lat_u]$, where l refers to the lower bound, and u refers to the upper bound. Suppose the resolution of the corresponding remote sensing image is $H \times W$, and all corresponding GPS trajectory points are projected onto the $H \times W$ gray-scale image through calculating the number of GPS points located in each pixel. Then the trajectory point density map is generated based on this method.

Trajectory Line Density Map Generation: The line segment is generated based on consecutive points in GPS trajectories. In the case of sparse GPS points, the line segments can help to recover the road network. Suppose the resolution of the corresponding remote sensing image is $H \times W$, and all corresponding GPS trajectory line segments are projected onto the $H \times W$ gray-scale image through calculating the number of line segments located in each pixel. Then the trajectory line density map is generated based on this method.

Binary Trajectory Point Map Generation: Suppose the resolution of the corresponding remote sensing image is $H \times W$, and all corresponding GPS trajectory points are projected onto the $H \times W$ gray-scale map. If there are GPS points projected into the pixel, then the pixel value is set to 1, otherwise, the pixel value is set to 0. Based on this method, the binary trajectory point map can be generated.

Binary Trajectory Line Map Generation: Suppose the resolution of the corre-

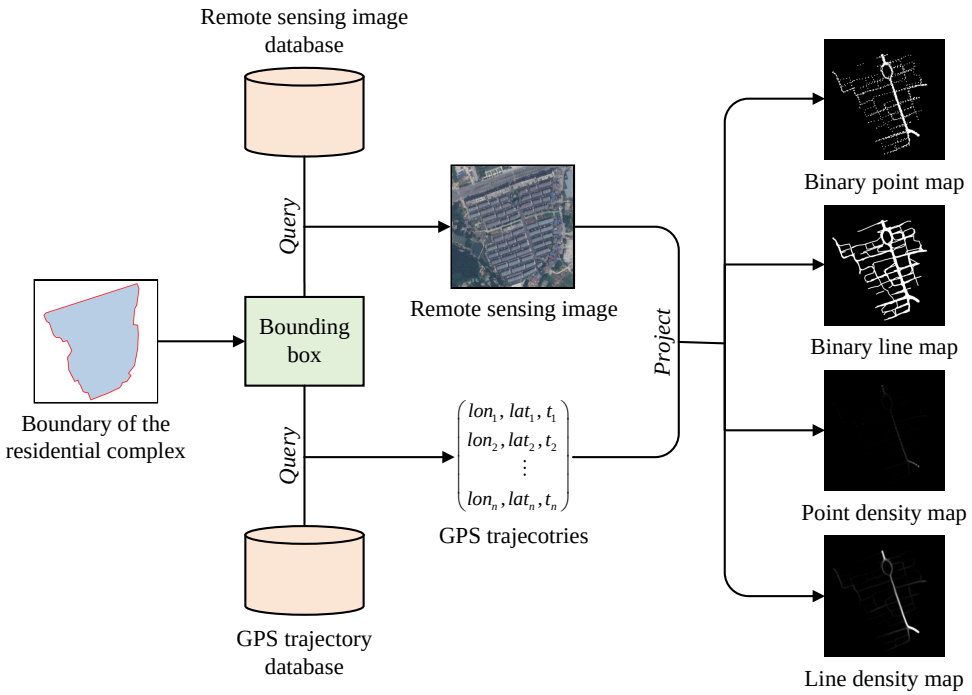


Figure 6.7: Illustration of GPS trajectory feature maps generation. Given a residential complex's boundary, we first get the bounding box of it, then query the remote sensing image and GPS trajectories from corresponding databases. Finally, we generate four 2D GPS trajectory feature maps projected from every pixel of the remote sensing image.

sponding remote sensing image is $H \times W$, and all corresponding GPS trajectory line segments are projected onto the $H \times W$ gray-scale map. If there are line segments projected into the pixel, then the pixel value is set to 1, otherwise, the pixel value is set to 0. Based on this method, the binary trajectory line map can be generated.

Image+Trajectory based Road Extraction: Given a remote sensing image I with a spatial resolution of $H \times W$, and the corresponding GPS trajectory feature maps (i.e., trajectory point density map T_{pd}, trajectory line density map T_{ld}, binary trajectory point map T_{pb}, and binary trajectory line map T_{lb}), the problem of the interior road extraction leveraging remote sensing images and GPS trajectories can be formulated as:

$$M = \mathcal{F}\left(\{I, T_{pd}, T_{ld}, T_{pb}, T_{lb}\}, \boldsymbol{\theta}\right), \tag{6.7}$$

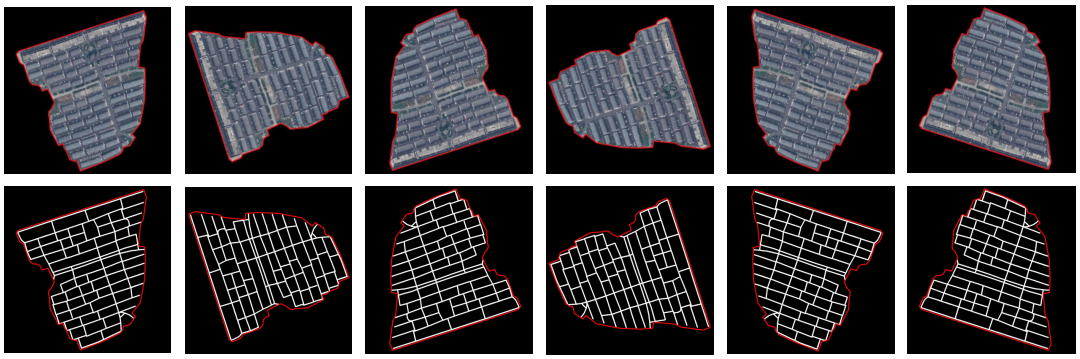
where $\boldsymbol{\theta}$ denotes the whole learnable parameters and $\mathcal{F}(\cdot)$ refers to a mapping function.

6.3.3 Data Augmentation

As a data-driven technology, the deep neural network is built on mass data to train the model with excellent performance. Image data augmentation is very important to ensure the invariance and robustness of the model, and avoid over-fitting. After applying data augmentation, the network can see a different set of images in each iteration. In order to keep the complete interior road network in every input images, only operations of rotation and flip are applied for the data augmentation. For each group of input feature maps and ground truth binary images, they are rotated with different angles (i.e., 90 degrees, 180 degrees, and 270 degrees), flipped horizontally and vertically, which are partially shown in Fig. 6.8. Finally, the amount of training data has been increased six times.

6.3.4 Experimental Settings

In this chapter, the input data dimension of remote sensing images is $640 \times 640 \times 3$, and the dimension of GPS trajectory feature maps is $640 \times 640 \times 4$, as shown in Fig. 6.3. The DF-DRUNet model and baseline models (i.e., U-Net [97], Res-U-Net [112], SegNet [103], DeepLabV3+ [100], DeConvNet [101], LinkNet [182], D-LinkNet [114], FuseNet [183], V-FuseNet [173], and DeepDualMapper [135]) are all implemented using Keras, and trained based on Nvidia Tesla M60 GPUs.



(a) (b) (c) (d) (e) (f)

Figure 6.8: Data augmentation. (a) The original remote sensing image and the corresponding ground truth image with the residential complex's boundary. (b) 90 degrees rotation. (c) 180 degrees rotation. (d) 270 degrees rotation. (e) Horizontal flip. (f) Vertical flip.

To compare with the repeated training's results, all hyperparameters are optimal. In the training process, Adam optimizer is utilized and we set the learning rate with 10^{-4}. The mini batch is set with 8, and all models are trained with 50 epochs.

6.3.5 Evaluation Metrics

Given an $H \times W$ probability map M, we can get a predicted interior road map $M_p \in \mathbb{R}^{H \times W}$. Then $M(x, y) \in [0, 1]$ refers to the probability value of the pixel (x, y) and $M_p(x, y) \in \{0, 1\}$ refers to the predicted value of the pixel (x, y). Similar to [40], if the value of $M(x, y)$ is greater than or equal to 0.5, then $M_p(x, y)$ is set to 1. Accordingly, if the value of $M(x, y)$ is less than 0.5, then $M_p(x, y)$ is set to 0. To evaluate the performance of our proposed method and other baseline methods, we introduce precision, recall, F1-score, and Intersection over Union (IoU) score as the evaluation metrics. Based on the predicted map M_p and its corresponding ground truth map M_g, the four evaluation metrics can be computed by Equations (6.8) to (6.11):

$$\text{Precision} = \frac{|M_p \cap M_g|}{|M_p|}, \tag{6.8}$$

$$\text{Recall} = \frac{|M_p \cap M_g|}{|M_g|}, \tag{6.9}$$

$$\begin{aligned} \text{F1-score} &= \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \\ &= \frac{2 \cdot |M_p \cap M_g|}{|M_p| \cdot |M_g|}, \end{aligned} \tag{6.10}$$

$$\text{IoU} = \frac{|M_p \cap M_g|}{|M_p \cup M_g|}, \tag{6.11}$$

where $|M_p|$ denotes the number of pixels in M_p with a predicted value of 1, and $|M_g|$ denotes the number of pixels in M_g with a ground truth value of 1. $|M_p \cap M_g|$ represents the number of pixels in the intersection of M_p and M_g. $|M_p \cup M_g|$ represents the number of pixels in the union of M_p and M_g.

6.3.6 Experimental Results

In this chapter, our experiments are divided into the following stages. The first stage is the data pre-processing and the generation of input feature maps. The second stage is to extract the interior road network based on our proposed method by fusing remote sensing images and GPS trajectories. The third stage is to post-process interior road regions based on noise removal, skeleton extraction, topology construction, vectorization and smoothing. In semantic segmentation task, F1-score

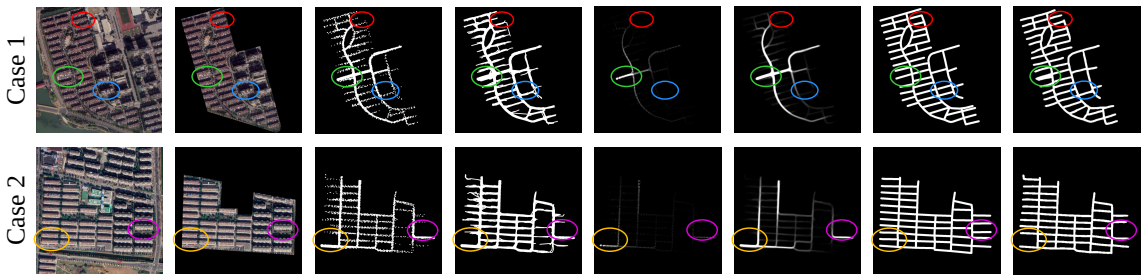


Figure 6.9: Results of the interior road extraction on the test dataset. (a) Original remote sensing images. (b) Remote sensing images within residential complexes. (c) Binary point maps of GPS trajectories. (d) Binary line maps of GPS trajectories. (e) Point density maps of GPS trajectories. (f) Line density maps of GPS trajectories. (g) Binary images of ground truth. (h) Binary images of predicted results.

and IoU are two critical metrics for evaluating the model's performance. The DF-DRUNet model achieves a good performance of F1-score with 85.11% and IoU with 74.26%.

Fig. 6.9 shows two cases of the interior road extraction results from the proposed model. In this figure, we use 8 columns to visualize the multi-source input images, the ground truth image and the predicted image. Fig. 6.9(a) shows the original remote sensing images and Fig. 6.9(b) converts the colour of remote sensing images outside the residential complex's boundary to black. From Fig. 6.9(c) to Fig. 6.9(f) are four GPS trajectory feature maps, i.e., binary trajectory point maps, binary trajectory line maps, trajectory point density maps, and trajectory line density maps. Fig. 6.9(g) illustrates the ground truth images of interior road regions and Fig. 6.9(h) is the predicted interior road regions based on the proposed model.

Based on the comparison among the input feature maps, the predicted images, and the ground truth images, some problems can still be found. In Case 1, within the red ellipses, the interior road regions are connected in the ground truth image, but the road links are disconnected in the predicted image. The reason why this happens is that there is no GPS trajectory information within the area of GPS trajectory

feature maps and the remote sensing image can not provide salient information in this area. In the green ellipses of Case 1, compared with the ground truth image, road regions are much wider in the predicted image. If we see the feature maps of GPS trajectories, we can see more trajectories in this area and also it is a main road in the residential complex. From the blue ellipses of Case 1, one interior road link is not detected in the predicted image due to limited information from input images. This issue also occurs in Case 2, in which the interior road links are not detected within the orange and purple ellipses due to limited information in the corresponding regions of the input data.

6.3.7 Comparison with Baseline Methods

In order to further evaluate the performance of the DF-DRUNet model, the proposed model is compared with various mainstream semantic segmentation models, including U-Net [97], Res-U-Net [112], SegNet [103], DeepLabV3+ [100], DeConvNet [101], LinkNet [182], D-LinkNet [114], FuseNet [183], V-FuseNet [173], and DeepDualMapper [135]. Specifically, the first seven baseline methods are re-implemented for multi-modal road extraction. For these methods, remote sensing images and GPS trajectory feature maps are directly concatenated as the input, and they are called the input fusion methods.

For methods of FuseNet and V-FuseNet, they consist of two independent models for feature extraction from remote sensing images and GPS trajectory feature maps, and the feature maps are fused together in decoder part, which are called the encoder fusion methods. Specifically, the model of DeepDualMapper also consists of two independent models for feature extraction from remote sensing images and GPS trajectory feature maps, then the feature maps are fused together by a gated fusion module in decoder part, which is called the decoder fusion method. Our proposed

Table 6.2: Quantitative comparison of semantic segmentation for the interior road extraction based on U-Net, Res-U-Net, SegNet, DeepLabV3+, DeConvNet, LinkNet, D-LinkNet, FuseNet, V-FuseNet, DeepDualMapper, and the proposed model in terms of Precision, Recall, F1-score, and IoU, where the bold values represent the best for corresponding metrics.

Method	Precision	Recall	F1-score	IoU
Input Fusion Methods				
U-Net [97]	84.94%	77.31%	80.80%	68.15%
Res-U-Net [112]	85.18%	77.90%	81.24%	68.77%
SegNet [103]	65.80%	79.13%	71.10%	56.18%
DeepLabV3+ [100]	**89.42%**	76.07%	81.97%	69.85%
DeConvNet [101]	84.97%	78.37%	81.37%	68.92%
LinkNet [182]	87.40%	77.35%	81.87%	69.56%
D-LinkNet [114]	86.07%	78.87%	82.13%	69.93%
Encoder Fusion Methods				
FuseNet [183]	86.01%	79.28%	82.34%	70.26%
V-FuseNet [173]	86.43%	79.85%	82.83%	71.03%
Decoder Fusion Methods				
DeepDualMapper [135]	86.02%	80.81%	83.18%	71.53%
Our Model	88.21%	**82.40%**	**85.11%**	**74.26%**

method is also a kind of decoder fusion method.

The performance of four evaluation metrics based on the proposed model and baseline models are illustrated in Table 6.2. On the whole, the proposed model performs the best of F1-score and IoU among all baseline models, which are 85.11% and 74.26%, respectively. The best value of recall is 82.02%, which is also achieved by our proposed model. While the best value of precision is 89.42%, which is achieved by DeepLabV3+. Among all input fusion methods, D-LinkNet achieves the best performance of F1-score and IoU, which are 82.13% and 69.93%, respectively. For encoder fusion methods, they have relatively better performance of F1-score and IoU compared with input fusion methods. Among all baseline models, DeepDualMapper achieves the best performance of F1-score and IoU, which are 83.18% and 71.03%,

respectively. Compared with other baseline models, SegNet has relatively worse performance of both F1-score and IoU, which is due to its low precision.

In conclusion, our proposed model is proved to be very effective in semantic segmentation of interior road extraction by comparing with other baseline models. As illustrated in Table 6.2, our model performs the best of F1-score and IoU among all baseline models.

6.4 Chapter Summary

In this chapter, we investigated the challenging task to extract the interior road network leveraging remote sensing images and GPS trajectories. To this end, we proposed a decoder fusion model called DF-DRUNet, which fuses remote sensing images and GPS trajectories more seamlessly. It contains two independent dilated Res-U-Net models with each taking remote sensing images and GPS trajectories as input modalities respectively. For the decoder fusion part, we fused decoders from two modalities based on a dual fusion module, which can help to learn the modality selection from these two modalities. We have demonstrated that the DF-DRUNet model is very effective based on the comprehensive numerical experiments, and the DF-DRUNet model achieves the best performance of F1-score (85.11%) and IoU (74.26%) among all baseline models.

Chapter 7

Conclusion and Future Directions

Spatio-temporal data mining is very important due to the availability of multiple sources of geographical and timestamped data that can be mined to solve many real-world problems in different applications. Spatio-temporal data mining aims to detect relationships and patterns that are invisible in spatio-temporal datasets. In this regard, this thesis investigates four efficient techniques in different scenarios for spatio-temporal data mining that take advantage of multi-source geospatial data to overcome the limitations of traditional data mining methods. A detailed literature review on the latest advances for spatio-temporal data mining is provided in Chapter 2. The detailed studies of this thesis are reported in Chapters 3−6. In this chapter, we provide a brief summarization of our research in Section 7.1 and some possible future directions in Section 7.2.

7.1 Conclusions

In this thesis, we investigate the extractions of different levels of spatial objects, namely, point-level, line-level, and polygon-level, which are digital representations of the real world. In Chapters 3−6, four different approaches have been developed for the extractions of spatial objects based on multi-source geospatial data. And these methods demonstrated that the exploitation of spatio-temporal data mining from multi-source geospatial data can improve the extraction accuracy when compared to traditional data mining methods. In Chapter 3, we proposed a method for geolocation prediction of tweets without geo-tags using Twitter data, gazetteer, and digital boundaries of the US. In Chapter 4, we developed an optimization model for simultaneous detection of multiple AOIs from historical OFD order data and road network data. In Chapter 5, we proposed a multi-task Res-U-Net model with attention mechanism for POI detection of high-rise buildings using remote sensing images and historical OFD order data. In Chapter 6, we developed a novel decoder fusion model based on dilated Res-U-Net for interior road extraction leveraging remote sensing images and GPS trajectory data. As can be observed from the experimental results, the proposed methods can achieve higher performance and overcome the limitations of traditional data mining methods.

Below are the summarized descriptions of achievements, contributions, and limitations of the four approaches.

7.1.1 Geolocation Inference Using Twitter Data

We proposed a multi-elemental geolocation inference method based on Twitter data, gazetteer, and digital boundaries of the US. This method has fully used all potential location-related attributes to predict tweets' geolocation. When the area threshold

of the bounding box is set to 10,000 km^2, the best model can successfully predict the geolocation of 90.8% of COVID-19 related tweets with the mean error distance of 4.824 km and the median error distance of 3.233 km. This method achieves the best performance compared with previous methods.

The contributions of this study can be summarized as follows. (1) Potential location-related attributes of the tweet's metadata are explored, and location entities are extracted via NER techniques. (2) We used three geographic coordinate datasets of counties to predict geolocation, and the proposed models are built according to different priorities of location-related attributes. (3) The proposed method enhances the granularity of geographic information of tweets and makes the surveillance of COVID-19 effective and efficient.

The limitations of this study can be summarized as follows. (1) The library of NER is limited and does not contain every county's name, which results in some useful information being filtered out. (2) Even though the distance threshold is introduced to reduce the interference caused by duplicate county names, there still exist counties with the same name located in the same bounding box. (3) Several location entities can be extracted based on NER in some cases, but only the first location entity that meets the criteria is chosen in this study, even though there is a possibility that the real location-related information does not always appear in the first place.

7.1.2 Detection of Multi-AOIs Using OFD Data

We proposed a novel optimization model to detect multi-AOIs simultaneously based on historical OFD order data and road network data. We also introduced geohash technique to improve computational performance and used two algorithms to fine-tune the detected spatial boundaries based on road network. The results of our experiments suggest that the detection results are promising, and our model achieves

the best average F1-score of the whole single-AOI detection models.

The contributions of this study can be summarized as follows. (1) By accounting for the spatial dependency among neighbouring AOIs, we ensured that our approach can produce AOI boundaries that are consistent with each other. (2) We formulated the problem as a BILP model which can be efficiently solved by standard branch-and-bound procedures. (3) Using the optimization model in the dataset collected from Meituan platform, the experimental results show that our model identifies Multi-AOIs and improves the average F1-score from 0.847 to 0.894 and achieves the best average F1-score among all single-AOI detection methods.

The limitations of this study can be summarized as follows. (1) We used only two types of algorithms to create candidate spatial boundaries. To improve the detection performance, we should add more promising algorithms in the single-AOI detection step. (2) A road network cannot separate all AOIs since some AOIs are separated by rivers, walls, or other man-made-barriers. (3) We used only GPS data for AOI detection, while other data sources can also provide some complementary information for AOI detection.

7.1.3 POI Detection Using Remote Sensing Images

We proposed a multi-task Res-U-Net model with attention mechanism to increase the performance of semantic segmentation of the building roofs and the whole building shapes, and applied a sequence of post-processing methods and the offset vector method to detect building footprints of high-rise buildings from remote sensing images. Based on historical order dataset from OFD platform known as Meituan platform, we also identified POI names of the detected building footprints. After designing the attention module, the ResNet block and the multi-task strategy in the structure of the network, the capability to enhance the model's performance and

optimize boundaries of segmentation results can be achieved. Compared with the existing models, our model achieved the best performance in terms of both F1-score and IoU. The results show that the attention mechanism, the ResNet block and the multi-task strategy could effectively increase the amount of information and improve the predictive capability of the model.

The contributions of this study can be summarized as follows. (1) We proposed a novel decoder fusion model with the DFM unit based on two independent dilated Res-U-Net models for semantic segmentation of the interior road from both remote sensing images and GPS trajectories. The DF-DRUNet model achieves the best performance of F1-score and IoU among all baseline models. (2) A novel DFM unit is designed to help to learn the modality selection from both the two modalities of remote sensing images and GPS trajectories. (3) Most of existing studies extracted road network outside residential complexes, while our study concentrated on interior road network extraction within residential complexes from multi-source data.

The limitations of this study can be summarized as follows. (1) Even though the proposed model works excellent as a supervised model, it relies on a set of manually labeled data. (2) The offset vector method cannot be applied to those buildings where the shapes of the building roofs and the building footprints are very different, such as buildings with pointed roofs and special form buildings. (3) Name parsing of POIs is based on a statistics method, which may result in multiple building footprints having the same name.

7.1.4 Road Extraction Using Multi-Source Data

We proposed a novel decoder fusion model called DF-DRUNet to extract interior road network within residential complexes using remote sensing images and GPS trajectories. The DF-DRUNet model is built on two components: First, two inde-

pendent dilated Res-U-Net models with each taking remote sensing images and GPS trajectories as input modalities respectively. Second, for the decoder fusion part, we fused decoders from two modalities based on a dual fusion module, which can help to learn the modality selection from these two modalities. We have demonstrated that the DF-DRUNet model is very effective based on comprehensive numerical experiments, and the DF-DRUNet model achieves the best performance of F1-score (85.11%) and IoU (74.26%) among all baseline models.

The contributions of this study can be summarized as follows. (1) We proposed a novel decoder fusion model with the DFM unit based on two independent dilated Res-U-Net models for semantic segmentation of the interior road from both remote sensing images and GPS trajectories. The DF-DRUNet model achieves the best performance of F1-score and IoU among all baseline models. (2) A novel DFM unit is designed to help to learn the modality selection from both the two modalities of remote sensing images and GPS trajectories. (3) Most of existing studies extracted road network outside residential complexes, while our study concentrated on interior road network extraction within residential complexes from multi-source data.

The limitations of this study can be summarized as follows. (1) In this study, we attempted the fusion of remote sensing images and GPS trajectories, while more related data sources can be added into this study. (2) Regions without GPS trajectory information and no salient information of remote sensing images cannot be well extracted. (3) Although the DF-DRUNet performs well as a supervised model, it relies on a large amount of manually labeled data, which is difficult to obtain.

7.2 Future Directions

Future directions would focus on developing new modelling and visualization meth-

ods that can integrate multiple spatio-temporal data mining tasks to solve more complicated scenarios. In this thesis, we described four different spatio-temporal data mining problems in different scenarios. Below are the summarized descriptions of future directions of different tasks.

For the study of geolocation inference using Twitter data (Chapter 3), the proposed method can also be applied to other emergency datasets, e.g., bushfires, typhoons, and earthquakes. When computing the average lon-lat coordinates of geo-tagged tweets located in a county, different weights can be added to each tweet. In addition, techniques such as natural language processing and deep learning models can strengthen the text analysis and promote the development of this research field. More importantly, the proposed models can provide inspirations for other related research and can be used in other Twitter related event studies.

For the study of simultaneous detection of multi-AOIs (Chapter 4), we used two types of algorithms to create candidate spatial boundaries. In the single-AOI detection step, we can add more promising algorithms such as P-DBSCAN, C-DBSCAN, M-DBSCAN and H-DBSCAN, which may help to improve the detection performance. Moreover, collecting more information of lower-level road network can increase the possibility of further region partition. Since some AOIs are separated by rivers, walls, or other man-made barriers, adding these networks can efficiently complement the road network. A possible extension of this study is to add other data sources, such as remote sensing data which can be used to perform image recognition of AOIs.

For the study of POI detection of high-rise buildings using remote sensing images (Chapter 5) and interior road extraction using multi-source data (Chapter 6), even though our proposed models work well as a supervised model, it relies on a massive amount of manually labelled data. Further studies are required to reduce the manual annotation. Potential directions for investigation include semi-supervised

semantic segmentation for adversarial learning, which can use unlabelled data to generate self-learning content signals to divide the network. Building footprint extraction also has been broadly studied in the field of photogrammetry, therefore, more related baseline models can be added in future research. For the study of interior road extraction using multi-source data, our proposed approach generates four GPS trajectory feature maps for the pre-processing of GPS trajectories, but we only considered only the information of GPS points and line segments. Therefore, we plan to add the information of speed and direction for input feature map generation and explore whether this strategy can further improve the model's performance. Moreover, we plan to use KDE to generate density feature maps. KDE could smooth out the randomness of where GPS points are recorded due to sampling rate or GPS error etc. About baseline models, more image processing-based approaches and other clustering-based approaches can be added in a future study. To further scientific knowledge, we will continue to work towards this direction, and explore the hybrid model to produce significant semantic explanations.

The training procedure for any model is related to the training samples, and the main problem of spatio-temporal data mining is that sufficient training samples are not available in most cases. The fusion of multi-sourced data can be a potential direction for improving performance of spatio-temporal data mining.

Even though deep learning-based approaches have been broadly applied into spatio-temporal data mining, they are still in the early stage of development. Deep learning can be incorporated with other methods, such as photogrammetry models and graphical models, to achieve better performance of spatio-temporal data mining.

Appendix A

Proof of the offset vector method

This appendix presents the detailed proof procedure of the offset vector method. For the offset vector method, the hypothesis of this research is that the polygon of the building roof has the same shape as the polygon of the building footprint in the real world. Generally, the distance between the orbit of a satellite and the earth's surface is thousands of kilometers, while the height of a building is only tens of meters. Therefore, the shape of the building roof and the corresponding building footprint in the remote sensing image can be recognized as the same.

To prove the offset vector is correct, we suppose the building is a parallelepiped. When the building is projected onto a plane, its whole shape is shown in Figure A.1(a). This figure should be seen as a two-dimensional image, thus, $AA_1 \,//\, BB_1 \,//\, CC_1 \,//\, DD_1$, $AB \,//\, DC \,//\, D_1C_1 \,//\, A_1B_1$. In Figure A.1(b), the light red polygon (the polygon $ABCD$) represents the polygon of the building roof, and the light green polygon (the polygon $A_1B_1C_1D_1$) represents the polygon of the building footprint. In Figure A.1(c), the light purple polygon (the polygon $A_1B_1C_1CDA$) represents the polygon of the whole building shape. We need to prove that the centroid of the polygon $ABCD$, the centroid of the polygon $A_1B_1C_1D_1$, and the centroid of

the polygon $A_1B_1C_1CDA$ all lie on the same vector. Both the polygon $ABCD$ and the polygon $A_1B_1C_1D_1$ are parallelograms, therefore, the centroids are the intersection of the diagonals. Then we need to calculate the centroid of the polygon $A_1B_1C_1CDA$.

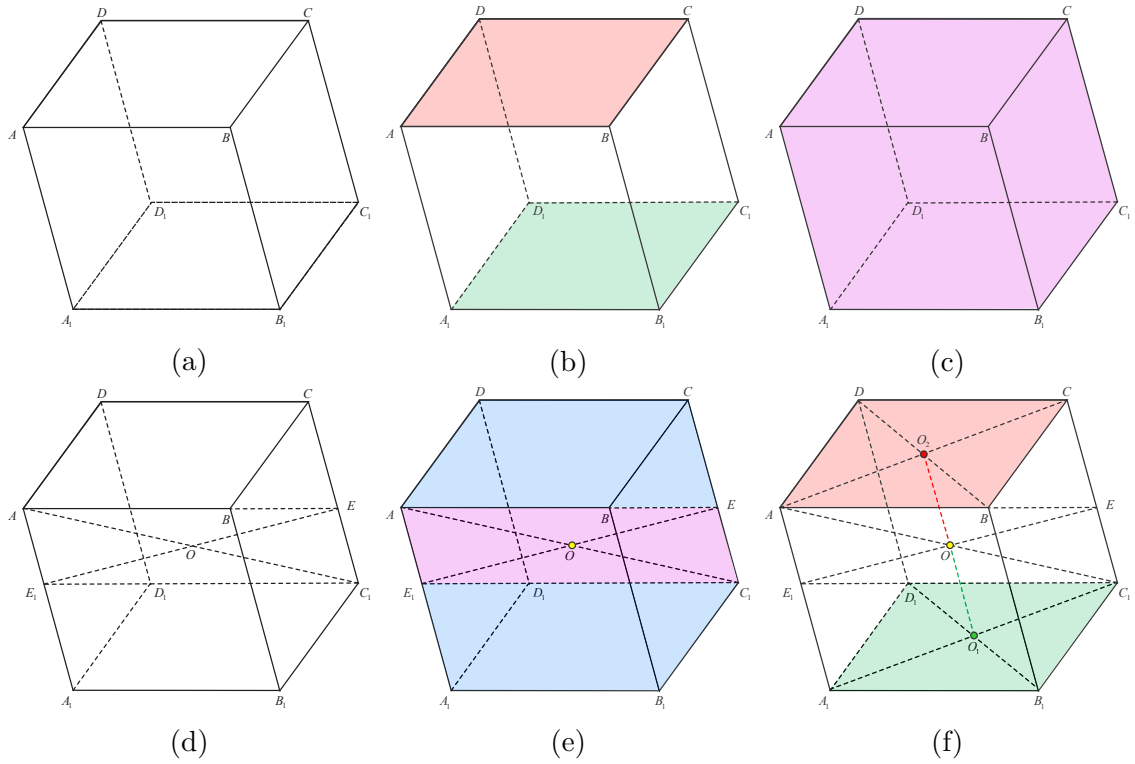


Figure A.1: An example of a building projected onto a plane. (a) The parallelepiped of the projected building. (b) Polygons of the building roof and the building footprint. (c) The polygon of the whole building shape. (d) Auxiliary lines are added for calculation. (e) Calculation of the centroid of the whole building shape. (f) Proof of centroids of the building roof, the building footprint and the whole building shape all lie on the same vector.

To calculate the centroid of the polygon $A_1B_1C_1CDA$, we need to add some auxiliary lines. In Figure A.1(d), extend the line segment AB to the line segment CC_1, and the intersection point is set to E. Then extend the line segment C_1D_1 to the line segment AA_1, and the intersection point is set to E_1. In Figure A.1(e), the polygon $A_1B_1C_1CDA$ is divided into three polygons with different colors, i.e., the polygon $AECD$, the polygon $A_1B_1C_1E_1$, and the polygon E_1C_1EA. It's easy to prove that

the polygon E_1C_1EA is a parallelogram, thus, the centroid of it is the intersection of its diagonals, which is the point O. And it's also easy to prove that the polygon $AECD$ is equivalent to the polygon $A_1B_1C_1E_1$, and they are centrosymmetric with respect to the point O. Therefore, the centroid of the polygon $AECD$ and the polygon $A_1B_1C_1E_1$ is the point O, and we can prove that the point O is the centroid of the polygon $A_1B_1C_1CDA$. In Figure A.1(f), the point O_2 is the centroid of the polygon $ABCD$, the point O_1 is the centroid of the polygon $A_1B_1C_1D_1$, and then we need to prove that the point O_2, the point O, and the point O_1 lie on the same line. Because the point O_2 is the midpoint of the line segment AC, the point O is the midpoint of the line segment E_1E, the point O_1 is the midpoint of the line segment A_1C_1, and $AA_1 // CC_1$, therefore, $OO_2 // CC_1$, $OO_1 // CC_1$, and the point O, the point O_1, the point O_2 all lie on the line segment O_1O_2. Finally, we prove that the centroid of the polygon $ABCD$, the centroid of the polygon $A_1B_1C_1D_1$, and the centroid of the polygon $A_1B_1C_1CDA$ all lie on the same vector.

For most of the residential buildings, their shapes are close to the cuboid. Therefore, the offset vector method can be applied to these buildings.

References

[1] L. Liu, Z. Yang, G. Li, K. Wang, T. Chen, and L. Lin, "Aerial images meet crowdsourced trajectories: a new approach to robust road extraction," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[2] C. Tran, D. D. Vu, and W.-Y. Shin, "An improved approach for estimating social poi boundaries with textual attributes on social media," *Knowledge-Based Systems*, vol. 213, p. 106710, 2021.

[3] V. Srivastava, P. Tejaswin, L. Dhakad, M. Kumar, and A. Dani, "A geocoding framework powered by delivery data," in *Proceedings of the 28th International Conference on Advances in Geographic Information Systems*, 2020, pp. 568–577.

[4] A. Hamdi, K. Shaban, A. Erradi, A. Mohamed, S. K. Rumi, and F. D. Salim, "Spatiotemporal data mining: a survey on challenges and open problems," *Artificial Intelligence Review*, vol. 55, no. 2, pp. 1441–1488, 2022.

[5] S. Wang, J. Cao, and P. Yu, "Deep learning for spatio-temporal data mining: A survey," *IEEE transactions on knowledge and data engineering*, 2020.

[6] R. Nandal, "Spatio-temporal database and its models: a review," *IOSR J. Comput. Eng*, vol. 11, no. 2, pp. 91–100, 2013.

[7] B. Mersch, T. Höllen, K. Zhao, C. Stachniss, and R. Roscher, "Maneuver-based trajectory prediction for self-driving cars using spatio-temporal convolutional networks," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 4888–4895.

[8] B. Li, J. Gao, S. Chen, S. Lim, and H. Jiang, "Poi detection of high-rise buildings using remote sensing images: A semantic segmentation method based on multi-task attention res-u-net," *IEEE Transactions on Geoscience and Remote Sensing*, 2022.

[9] J. Bracher and L. Held, "Endemic-epidemic models with discrete-time serial interval distributions for infectious disease prediction," *International Journal of Forecasting*, 2020.

[10] W. Wang, N. Yang, Y. Zhang, F. Wang, T. Cao, and P. Eklund, "A review of road extraction from remote sensing images," *Journal of Traffic and Transportation Engineering*, vol. 3, no. 3, pp. 271–282, 2016.

[11] G. Atluri, A. Karpatne, and V. Kumar, "Spatio-temporal data mining: A survey of problems and methods," *ACM Computing Surveys (CSUR)*, vol. 51, no. 4, pp. 1–41, 2018.

[12] V. M. Prieto, S. Matos, M. Alvarez, F. Cacheda, and J. L. Oliveira, "Twitter: a good place to detect health conditions," *PLOS One*, vol. 9, no. 1, pp. 1–11, 2014.

[13] M. Paul and M. Dredze, "You are what you tweet: Analyzing twitter for public health," in *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 5, no. 1, 2011.

[14] E. Steiger, J. P. De Albuquerque, and A. Zipf, "An advanced systematic literature review on spatiotemporal analyses of twitter data," *Transactions in GIS*, vol. 19, no. 6, pp. 809–834, 2015.

[15] A. Crooks, A. Croitoru, A. Stefanidis, and J. Radzikowski, "#earthquake: Twitter as a distributed sensor system," *Transactions in GIS*, vol. 17, no. 1, pp. 124–147, 2013.

[16] L. Sinnenberg, A. M. Buttenheim, K. Padrez, C. Mancheno, L. Ungar, and R. M. Merchant, "Twitter as a tool for health research: a systematic review," *American Journal of Public Health*, vol. 107, no. 1, pp. e1–e8, 2017.

[17] O. Ajao, J. Hong, and W. Liu, "A survey of location inference techniques on twitter," *Journal of Information Science*, vol. 41, no. 6, pp. 855–864, 2015.

[18] C.-Y. Huang, H. Tong, J. He, and R. Maciejewski, "Location prediction for tweets," *Frontiers in Big Data*, vol. 2, p. 5, 2019.

[19] F. Laylavi, A. Rajabifard, and M. Kalantari, "A multi-element approach to location inference of twitter: A case for emergency response," *ISPRS International Journal of Geo-Information*, vol. 5, no. 5, p. 56, 2016.

[20] C. Allen, M.-H. Tsou, A. Aslam, A. Nagel, and J.-M. Gawron, "Applying gis and machine learning methods to twitter data for multiscale surveillance of influenza," *PLOS One*, vol. 11, no. 7, pp. 1–10, 2016.

[21] Y. Gao, S. Wang, A. Padmanabhan, J. Yin, and G. Cao, "Mapping spatiotemporal patterns of events using social media: a case study of influenza trends," *International Journal of Geographical Information Science*, vol. 32, no. 3, pp. 425–449, 2018.

[22] J. Zheng, L. Wang, S. Wang, Y. Liang, and J. Pan, "Solving two-stage stochastic route-planning problem in milliseconds via end-to-end deep learning," *Complex & Intelligent Systems*, vol. 7, no. 3, pp. 1207–1222, 2021.

[23] J. Kranjec, "Online food delivery market to hit $151.5b in revenue and 1.6b users in 2021, a 10% jump in a year," 2021. [Online]. Available: https://bit.ly/3v1jU3d

[24] C. Li, M. Mirosa, and P. Bremer, "Review of online food delivery platforms and their impacts on sustainability," *Sustainability*, vol. 12, no. 14, p. 5528, 2020.

[25] Meituan, "Meituan announces financial results for the year ended december 31, 2020," 2021. [Online]. Available: https://prn.to/3yWcWPa

[26] G. McKenzie, K. Janowicz, S. Gao, J.-A. Yang, and Y. Hu, "Poi pulse: A multi-granular, semantic signature–based information observatory for the interactive visualization of big geosocial data," *Cartographica: The International Journal for Geographic Information and Geovisualization*, vol. 50, no. 2, pp. 71–85, 2015.

[27] L. Zhang and S. Wang, "Region-of-interest extraction based on local–global contrast analysis and intra-spectrum information distribution estimation for remote sensing images," *Remote Sensing*, vol. 9, no. 6, p. 597, 2017.

[28] Y. Hu, S. Gao, K. Janowicz, B. Yu, W. Li, and S. Prasad, "Extracting and understanding urban areas of interest using geotagged photos," *Computers, Environment and Urban Systems*, vol. 54, pp. 240–254, 2015.

[29] E. Spyrou and P. Mylonas, "Analyzing flickr metadata to extract location-based information and semantically organize its photo content," *Neurocomputing*, vol. 172, pp. 114–133, 2016.

[30] J. Sun, T. Kinoue, and Q. Ma, "A city adaptive clustering framework for discovering pois with different granularities," in *International Conference on Database and Expert Systems Applications*. Springer, 2020, pp. 425–434.

[31] L. Belcastro, F. Marozzo, D. Talia, and P. Trunfio, "G-roi: Automatic region-of-interest detection driven by geotagged social media data," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 12, no. 3, pp. 1–22, 2018.

[32] X. Liu, Q. Huang, and S. Gao, "Exploring the uncertainty of activity zone detection using digital footprints with multi-scaled dbscan," *International Journal of Geographical Information Science*, vol. 33, no. 6, pp. 1196–1223, 2019.

[33] B. Devkota, H. Miyazaki, A. Witayangkurn, and S. M. Kim, "Using volunteered geographic information and nighttime light remote sensing data to identify tourism areas of interest," *Sustainability*, vol. 11, no. 17, p. 4718, 2019.

[34] X. Zhang, Y. Sun, A. Zheng, and Y. Wang, "A new approach to refining land use types: Predicting point-of-interest categories using weibo check-in data," *ISPRS International Journal of Geo-Information*, vol. 9, no. 2, p. 124, 2020.

[35] W. Li, C. He, J. Fang, J. Zheng, H. Fu, and L. Yu, "Semantic segmentation-based building footprint extraction using very high-resolution satellite images and multi-source gis data," *Remote Sensing*, vol. 11, no. 4, p. 403, 2019.

[36] K. Zhao, M. Kamran, and G. Sohn, "Boundary regularized building footprint extraction from satellite images using deep neural network," *arXiv preprint arXiv:2006.13176*, 2020.

[37] A. Ziaee, R. Dehbozorgi, and M. Döller, "A novel adaptive deep network for building footprint segmentation," *arXiv preprint arXiv:2103.00286*, 2021.

[38] J. Biagioni and J. Eriksson, "Map inference in the face of noise and disparity," in *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*, 2012, pp. 79–88.

[39] S. Ruan, C. Long, J. Bao, C. Li, Z. Yu, R. Li, Y. Liang, T. He, and Y. Zheng, "Learning to generate maps from trajectories," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 01, 2020, pp. 890–897.

[40] T. Sun, Z. Di, P. Che, C. Liu, and Y. Wang, "Leveraging crowdsourced gps data for road extraction from aerial imagery," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7509–7518.

[41] Y. Li, L. Xiang, C. Zhang, and H. Wu, "Fusing taxi trajectories and rs images to build road map via dcnn," *IEEE Access*, vol. 7, pp. 161 487–161 498, 2019.

[42] A. A. Alalwan, N. P. Rana, Y. K. Dwivedi, and R. Algharabat, "Social media in marketing: A review and analysis of the existing literature," *Telematics and Informatics*, vol. 34, no. 7, pp. 1177–1190, 2017.

[43] M. Ahlgren, "50+ twitter statistics & facts for 2022," 2022. [Online]. Available: https://www.websitehostingrating.com/twitter-statistics/

[44] W. Li, P. Serdyukov, A. P. de Vries, C. Eickhoff, and M. Larson, "The where in the tweet," in *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, 2011, pp. 2473–2476.

[45] Z. Cheng, J. Caverlee, and K. Lee, "A content-driven framework for geolocating microblog users," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 4, no. 1, pp. 1–27, 2013.

[46] B. Hecht, L. Hong, B. Suh, and E. H. Chi, "Tweets from justin bieber's heart: the dynamics of the location field in user profiles," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2011, pp. 237–246.

[47] K. Ryoo and S. Moon, "Inferring twitter user locations with 10 km accuracy," in *Proceedings of the 23rd International Conference on World Wide Web*, 2014, pp. 643–648.

[48] R. Priedhorsky, A. Culotta, and S. Y. Del Valle, "Inferring the origin locations of tweets with quantitative confidence," in *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing*, 2014, pp. 1523–1536.

[49] Z. Cheng, J. Caverlee, and K. Lee, "You are where you tweet: a content-based approach to geo-locating twitter users," in *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, 2010, pp. 759–768.

[50] S. Chandra, L. Khan, and F. B. Muhaya, "Estimating twitter user location using social interactions–a content based approach," in *2011 IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing*. IEEE, 2011, pp. 838–843.

[51] H.-w. Chang, D. Lee, M. Eltaher, and J. Lee, "@ phillies tweeting from philly? predicting twitter user locations with spatial word usage," in *2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*. IEEE, 2012, pp. 111–118.

[52] Y. Ikawa, M. Vukovic, J. Rogstadius, and A. Murakami, "Location-based insights from the social web," in *Proceedings of the 22nd International Conference on World Wide Web*, 2013, pp. 1013–1016.

[53] S. Abrol and L. Khan, "Tweethood: Agglomerative clustering on fuzzy k-closest friends with variable depth for location mining," in *2010 IEEE Second International Conference on Social Computing*. IEEE, 2010, pp. 153–160.

[54] L. Backstrom, E. Sun, and C. Marlow, "Find me if you can: improving geographical prediction with social and spatial proximity," in *Proceedings of the 19th International Conference on World Wide Web*, 2010, pp. 61–70.

[55] F. Bouillot, P. Poncelet, and M. Roche, "How and why exploit tweet's location information?" in *AGILE International Conference on Geographic Information Science*. Springer, 2012.

[56] J. Lingad, S. Karimi, and J. Yin, "Location extraction from disaster-related microblogs," in *Proceedings of the 22nd International Conference on World Wide Web*, 2013, pp. 1017–1020.

[57] R. Li, S. Wang, H. Deng, R. Wang, and K. C.-C. Chang, "Towards social user profiling: unified and discriminative influence model for inferring home locations," in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2012, pp. 1023–1031.

[58] Y. Takhteyev, A. Gruzd, and B. Wellman, "Geography of twitter networks," *Social Networks*, vol. 34, no. 1, pp. 73–81, 2012.

[59] C. Li and A. Sun, "Fine-grained location extraction from tweets with temporal awareness," in *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*, 2014, pp. 43–52.

[60] M. F. Goodchild, "Citizens as sensors: the world of volunteered geography," *GeoJournal*, vol. 69, no. 4, pp. 211–221, 2007.

[61] D. Sui, S. Elwood, and M. Goodchild, *Crowdsourcing Geographic Knowledge: Volunteered Geographic Information (VGI) in Theory and Practice*. Springer Science & Business Media, 2012.

[62] L. Li and M. F. Goodchild, "Constructing places from spatial footprints," in *Proceedings of the 1st ACM SIGSPATIAL International Workshop on Crowdsourced and Volunteered Geographic Information*, 2012, pp. 15–21.

[63] Z. Cheng, J. Caverlee, K. Lee, and D. Sui, "Exploring millions of footprints in location sharing services," in *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 5, 2011.

[64] H. Q. Vu, G. Li, R. Law, and B. H. Ye, "Exploring the travel behaviors of inbound tourists to hong kong using geotagged photos," *Tourism Management*, vol. 46, pp. 222–232, 2015.

[65] C.-L. Kuo, T.-C. Chan, I. Fan, A. Zipf *et al.*, "Efficient method for poi/roi discovery using flickr geotagged photos," *ISPRS International Journal of Geo-Information*, vol. 7, no. 3, p. 121, 2018.

[66] A. Altomare, E. Cesario, C. Comito, F. Marozzo, and D. Talia, "Trajectory pattern mining for urban computing in the cloud," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 2, pp. 586–599, 2016.

[67] D. J. Crandall, L. Backstrom, D. Huttenlocher, and J. Kleinberg, "Mapping the world's photos," in *Proceedings of the 18th International Conference on World Wide Web*. Citeseer, 2009, pp. 761–770.

[68] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, 2002.

[69] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *KDD*, vol. 96, 1996, pp. 226–231.

[70] S. Kisilevich, F. Mansmann, and D. Keim, "P-dbscan: A density based clustering algorithm for exploration and analysis of attractive areas using collections of geo-tagged photos," in *Proceedings of the 1st International Conference and Exhibition on Computing for Geospatial Research & Application*, 2010, pp. 1–4.

[71] C. Ruiz, M. Spiliopoulou, and E. Menasalvas, "C-dbscan: Density-based clustering with constraints," in *International Workshop on Rough Sets, Fuzzy Sets, Data Mining, and Granular-Soft Computing*. Springer, 2007, pp. 216–223.

[72] R. J. Campello, D. Moulavi, and J. Sander, "Density-based clustering based on hierarchical density estimates," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2013, pp. 160–172.

[73] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "Optics: Ordering points to identify the clustering structure," *ACM Sigmod Record*, vol. 28, no. 2, pp. 49–60, 1999.

[74] M. Ibrahim, "Extracting and mapping areas of interest from social media," Ph.D. dissertation, Wien, 2020.

[75] J. Liu, Z. Huang, L. Chen, H. T. Shen, and Z. Yan, "Discovering areas of interest with geo-tagged images and check-ins," in *Proceedings of the 20th ACM International Conference on Multimedia*, 2012, pp. 589–598.

[76] D. Laptev, A. Tikhonov, P. Serdyukov, and G. Gusev, "Parameter-free discovery and recommendation of areas-of-interest," in *Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2014, pp. 113–122.

[77] M. Shirai, M. Hirota, H. Ishikawa, and S. Yokoyama, "A method of area of interest and shooting spot detection using geo-tagged photographs," in *Pro-*

ceedings of the First ACM SIGSPATIAL International Workshop on Computational Models of Place, 2013, pp. 34–41.

[78] S. M. Mousavi, A. Harwood, S. Karunasekera, and M. Maghrebi, "Geometry of interest (goi): spatio-temporal destination extraction and partitioning in gps trajectory data," *Journal of Ambient Intelligence and Humanized Computing*, vol. 8, no. 3, pp. 419–434, 2017.

[79] A. Dubray, G. Derval, S. Nijssen, and P. Schaus, "Mining constrained regions of interest: An optimization approach," in *International Conference on Discovery Science*. Springer, 2020, pp. 630–644.

[80] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa, "The quickhull algorithm for convex hulls," *ACM Transactions on Mathematical Software (TOMS)*, vol. 22, no. 4, pp. 469–483, 1996.

[81] H. Edelsbrunner, D. Kirkpatrick, and R. Seidel, "On the shape of a set of points in the plane," *IEEE Transactions on Information Theory*, vol. 29, no. 4, pp. 551–559, 1983.

[82] M. Duckham, L. Kulik, M. Worboys, and A. Galton, "Efficient generation of simple polygons for characterizing the shape of a set of points in the plane," *Pattern Recognition*, vol. 41, no. 10, pp. 3224–3236, 2008.

[83] J. Hui, M. Du, X. Ye, Q. Qin, and J. Sui, "Effective building extraction from high-resolution remote sensing images with multitask driven deep neural network," *IEEE Geoscience and Remote Sensing Letters*, vol. 16, no. 5, pp. 786–790, 2018.

[84] Y. Sun, X. Zhang, X. Zhao, and Q. Xin, "Extracting building boundaries from high resolution optical images and lidar data by integrating the convolutional neural network and the active contour model," *Remote Sensing*, vol. 10, no. 9, p. 1459, 2018.

[85] L. Li, J. Liang, M. Weng, and H. Zhu, "A multiple-feature reuse network to extract buildings from remote sensing imagery," *Remote Sensing*, vol. 10, no. 9, p. 1350, 2018.

[86] E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez, "Convolutional neural networks for large-scale remote-sensing image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 2, pp. 645–657, 2016.

[87] M. Belgiu and L. Drăguţ, "Comparing supervised and unsupervised multiresolution segmentation approaches for extracting buildings from very high resolution imagery," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 96, pp. 67–75, 2014.

[88] R. Chen, X. Li, and J. Li, "Object-based features for house detection from rgb high-resolution images," *Remote Sensing*, vol. 10, no. 3, p. 451, 2018.

[89] A. O. Ok, C. Senaras, and B. Yuksel, "Automated detection of arbitrarily shaped buildings in complex environments from monocular vhr optical satellite imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 51, no. 3, pp. 1701–1717, 2012.

[90] P. Ding, Y. Zhang, W.-J. Deng, P. Jia, and A. Kuijper, "A light and faster regional convolutional neural network for object detection in optical remote sensing images," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 141, pp. 208–218, 2018.

[91] W. Li, R. Dong, H. Fu, and L. Yu, "Large-scale oil palm tree detection from high-resolution satellite images using two-stage convolutional neural networks," *Remote Sensing*, vol. 11, no. 1, p. 11, 2019.

[92] Y. Liu, Y. Zhong, F. Fei, Q. Zhu, and Q. Qin, "Scene classification based on a deep random-scale stretched convolutional neural network," *Remote Sensing*, vol. 10, no. 3, p. 444, 2018.

[93] B. Huang, B. Zhao, and Y. Song, "Urban land-use mapping using a deep convolutional neural network with high spatial resolution multispectral remote sensing imagery," *Remote Sensing of Environment*, vol. 214, pp. 73–86, 2018.

[94] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[95] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[96] M. Alam, J.-F. Wang, C. Guangpei, L. Yunrong, and Y. Chen, "Convolutional neural network for the semantic segmentation of remote sensing images," *Mobile Networks and Applications*, vol. 26, no. 1, pp. 200–215, 2021.

[97] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2015, pp. 234–241.

[98] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected crfs," *arXiv preprint arXiv:1412.7062*, 2014.

[99] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834–848, 2017.

[100] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 801–818.

[101] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1520–1528.

[102] J. Yang, B. Price, S. Cohen, H. Lee, and M.-H. Yang, "Object contour detection with a fully convolutional encoder-decoder network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 193–202.

[103] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[104] D. Marmanis, K. Schindler, J. D. Wegner, S. Galliani, M. Datcu, and U. Stilla, "Classification with an edge: Improving semantic image segmentation with boundary detection," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 135, pp. 158–172, 2018.

[105] Q. Shi, M. Liu, S. Li, X. Liu, F. Wang, and L. Zhang, "A deeply supervised attention metric-based network and an open aerial image dataset for remote sensing change detection," *IEEE Transactions on Geoscience and Remote Sensing*, 2021.

[106] D. He, Q. Shi, X. Liu, Y. Zhong, and L. Zhang, "Generating 2m fine-scale urban tree cover product over 34 metropolises in china based on deep context-aware sub-pixel mapping network," *International Journal of Applied Earth Observation and Geoinformation*, vol. 106, p. 102667, 2022.

[107] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint arXiv:1706.05587*, 2017.

[108] J. F. Mas and J. J. Flores, "The application of artificial neural networks to the analysis of remotely sensed data," *International Journal of Remote Sensing*, vol. 29, no. 3, pp. 617–663, 2008.

[109] M. Guo, H. Liu, Y. Xu, and Y. Huang, "Building extraction based on u-net with an attention block and multiple losses," *Remote Sensing*, vol. 12, no. 9, p. 1400, 2020.

[110] M. Sahu and A. Ohri, "Vector map generation from aerial imagery using deep learning," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2019.

[111] E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez, "Fully convolutional neural networks for remote sensing image classification," in *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. IEEE, 2016, pp. 5071–5074.

[112] Y. Xu, L. Wu, Z. Xie, and Z. Chen, "Building extraction in very high resolution remote sensing imagery using deep learning and guided filters," *Remote Sensing*, vol. 10, no. 1, p. 144, 2018.

[113] X. Chen, C. Qiu, W. Guo, A. Yu, X. Tong, and M. Schmitt, "Multiscale feature learning by transformer for building extraction from satellite images," *IEEE Geoscience and Remote Sensing Letters*, 2022.

[114] L. Zhou, C. Zhang, and M. Wu, "D-linknet: Linknet with pretrained encoder and dilated convolution for high resolution satellite imagery road extraction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 182–186.

[115] S. Hinz and A. Baumgartner, "Automatic extraction of urban road networks from multi-view aerial imagery," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 58, no. 1-2, pp. 83–98, 2003.

[116] P. Anil and S. Natarajan, "A novel approach using active contour model for semi-automatic road extraction from high resolution satellite imagery," in *2010 Second International Conference on Machine Learning and Computing*. IEEE, 2010, pp. 263–266.

[117] D. Chaudhuri, N. K. Kushwaha, and A. Samal, "Semi-automated road detection from high resolution satellite images by directional morphological enhancement and segmentation techniques," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 5, no. 5, pp. 1538–1544, 2012.

[118] G. Cheng, Y. Wang, S. Xu, H. Wang, S. Xiang, and C. Pan, "Automatic road detection and centerline extraction via cascaded end-to-end convolutional neural network," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 6, pp. 3322–3337, 2017.

[119] Z. Zhang, Q. Liu, and Y. Wang, "Road extraction by deep residual u-net," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 5, pp. 749–753, 2018.

[120] T. Alshaikhli, W. Liu, and Y. Maruyama, "Automated method of road extraction from aerial images using a deep convolutional neural network," *Applied Sciences*, vol. 9, no. 22, p. 4825, 2019.

[121] K. K. Eerapu, B. Ashwath, S. Lal, F. Dell'Acqua, and A. N. Dhan, "Dense refinement residual network for road extraction from aerial imagery data," *IEEE Access*, vol. 7, pp. 151 764–151 782, 2019.

[122] X. Zhang, W. Ma, C. Li, J. Wu, X. Tang, and L. Jiao, "Fully convolutional network-based ensemble method for road extraction from aerial images," *IEEE Geoscience and Remote Sensing Letters*, vol. 17, no. 10, pp. 1777–1781, 2019.

[123] S. Edelkamp and S. Schrödl, "Route planning and map inference with global positioning traces," in *Computer Science in Perspective*. Springer, 2003, pp. 128–151.

[124] C. Chen, C. Lu, Q. Huang, Q. Yang, D. Gunopulos, and L. Guibas, "City-scale map creation and updating using gps collections," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 1465–1474.

[125] R. Stanojevic, S. Abbar, S. Thirumuruganathan, S. Chawla, F. Filali, and A. Aleimat, "Robust road map inference through network alignment of trajectories," in *Proceedings of the 2018 SIAM International Conference on Data Mining*. SIAM, 2018, pp. 135–143.

[126] L. Cao and J. Krumm, "From gps traces to a routable road map," in *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2009, pp. 3–12.

[127] B. Niehoefer, R. Burda, C. Wietfeld, F. Bauer, and O. Lueert, "Gps community map generation for enhanced routing methods based on trace-collection by mobile phones," in *2009 First International Conference on Advances in Satellite and Space Communications*. IEEE, 2009, pp. 156–161.

[128] S. Wang, Y. Wang, and Y. Li, "Efficient map reconstruction and augmentation via topological methods," in *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2015, pp. 1–10.

[129] S. Clode, P. J. Kootsookos, and F. Rottensteiner, "The automatic extraction of roads from lidar data." ISPRS, 2004.

[130] Z. Hui, Y. Hu, S. Jin, and Y. Z. Yevenyo, "Road centerline extraction from airborne lidar point cloud based on hierarchical fusion and optimization," *IS-*

PRS Journal of Photogrammetry and Remote Sensing, vol. 118, pp. 22–36, 2016.

[131] W. Zhang, "Lidar-based road and road-edge detection," in *2010 IEEE Intelligent Vehicles Symposium*. IEEE, 2010, pp. 845–848.

[132] X. Hu, Y. Li, J. Shan, J. Zhang, and Y. Zhang, "Road centerline extraction in complex urban scenes from lidar data based on multiple features," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 11, pp. 7448–7456, 2014.

[133] Y. Li, L. Ma, Z. Zhong, F. Liu, M. A. Chapman, D. Cao, and J. Li, "Deep learning for lidar point clouds in autonomous driving: A review," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 8, pp. 3412–3432, 2020.

[134] X. Hu, C. V. Tao, and Y. Hu, "Automatic road extraction from dense urban area by integrated processing of high resolution imagery and lidar data," *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 35, no. Part B3, pp. 288–292, 2004.

[135] H. Wu, H. Zhang, X. Zhang, W. Sun, B. Zheng, and Y. Jiang, "Deepdualmapper: A gated fusion network for automatic map extraction using aerial images and trajectories," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 01, 2020, pp. 1037–1045.

[136] Y. Yin, A. Tran, Y. Zhang, W. Hu, G. Wang, J. Varadarajan, R. Zimmermann, and S.-K. Ng, "Multimodal fusion of satellite images and crowdsourced gps traces for robust road attribute detection," in *Proceedings of the 29th International Conference on Advances in Geographic Information Systems*, 2021, pp. 107–116.

[137] L. Singh, S. Bansal, L. Bode, C. Budak, G. Chi, K. Kawintiranon, C. Padden, R. Vanarsdall, E. Vraga, and Y. Wang, "A first look at covid-19 information and misinformation sharing on twitter," *arXiv preprint arXiv:2003.13907*, 2020.

[138] J. M. Banda, R. Tekumalla, G. Wang, J. Yu, T. Liu, Y. Ding, E. Artemova, E. Tutubalina, and G. Chowell, "A large-scale covid-19 twitter chatter dataset for open scientific research—an international collaboration," *Epidemiologia*, vol. 2, no. 3, pp. 315–324, 2021.

[139] Worldometers, "Coronavirus update (live): 565,606,477 cases and 6,382,616 death from covid-19 virus," 2020. [Online]. Available: https://www.worldometers.info/coronavirus/

[140] Y. Lin, "10 twitter statistics every marketer should know in 2019," 2020. [Online]. Available: https://au.oberlo.com/blog/twitter-statistics

[141] A. Rosen, "Tweeting made easier," 2020. [Online]. Available: https://bit.ly/3yycGoc

[142] L. Tagliaferri, "An introduction to json," 2020. [Online]. Available: https://www.digitalocean.com/community/tutorials/an-introduction-to-json

[143] B. Li, Z. Chen, and S. Lim, "Geolocation prediction from tweets: A case study of influenza-like illness in australia," in *GISTAM*, 2020, pp. 160–167.

[144] J. P. Singh, Y. K. Dwivedi, N. P. Rana, A. Kumar, and K. K. Kapoor, "Event classification and location prediction from tweets during disasters," *Annals of Operations Research*, vol. 283, no. 1, pp. 737–757, 2019.

[145] Microsoft, "Regular expression language - quick reference," 2020. [Online]. Available: https://docs.microsoft.com/en-us/dotnet/standard/base-types/regular-expression-language-quick-reference

[146] L. Marujo, W. Ling, I. Trancoso, C. Dyer, A. W. Black, A. Gershman, D. M. de Matos, J. P. Neto, and J. G. Carbonell, "Automatic keyword extraction on twitter," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, 2015, pp. 637–643.

[147] R. Kouzy, J. Abi Jaoude, A. Kraitem, M. B. El Alam, B. Karam, E. Adib, J. Zarka, C. Traboulsi, E. W. Akl, and K. Baddour, "Coronavirus goes viral: quantifying the covid-19 misinformation epidemic on twitter," *Cureus*, vol. 12, no. 3, 2020.

[148] E. Chen, K. Lerman, E. Ferrara *et al.*, "Tracking social media discourse about the covid-19 pandemic: Development of a public coronavirus twitter data set," *JMIR Public Health and Surveillance*, vol. 6, no. 2, p. e19273, 2020.

[149] U. S. C. Bureau, "The national counties gazetteer file," 2020. [Online]. Available: https://www.census.gov/geographies/reference-files/time-series/geo/gazetteer-files.html

[150] E. S. R. Institute, "Usa counties," 2020. [Online]. Available: https://www.arcgis.com/home/item.html?id=a00d6b6149b34ed3b833e10fb72ef47b

[151] IBM, "Python tutorial of ibm ilog cplex 20.1.0," 2021. [Online]. Available: https://www.ibm.com/docs/en/icos/20.1.0?topic=tutorials-python-tutorial

[152] R. Windsor, "Meituan-dianping: Finally, a chinese tech ipo with some value," 2018. [Online]. Available: https://bit.ly/3uCpys0

[153] N. J. Yuan, Y. Zheng, and X. Xie, "Segmentation of urban areas using road networks," *Microsoft, Albuquerque, NM, USA, Tech. Rep. MSR-TR-2012-65*, July 2012.

[154] T. Fan, N. Guo, and Y. Ren, "Consumer clusters detection with geo-tagged social network data using dbscan algorithm: a case study of the pearl river delta in china," *GeoJournal*, vol. 86, no. 1, pp. 317–337, 2021.

[155] P. Newson and J. Krumm, "Hidden markov map matching through noise and sparseness," in *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2009, pp. 336–343.

[156] S. Shrestha and L. Vanneschi, "Improved fully convolutional network with conditional random fields for building extraction," *Remote Sensing*, vol. 10, no. 7, p. 1135, 2018.

[157] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus, "Deconvolutional networks," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 2010, pp. 2528–2535.

[158] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[159] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning*. PMLR, 2015, pp. 448–456.

[160] O. Oktay, J. Schlemper, L. L. Folgoc, M. Lee, M. Heinrich, K. Misawa, K. Mori, S. McDonagh, N. Y. Hammerla, B. Kainz *et al.*, "Attention u-net: Learning where to look for the pancreas," *arXiv preprint arXiv:1804.03999*, 2018.

[161] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang, "Residual attention network for image classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3156–3164.

[162] R. Bandara, "Image segmentation using unsupervised watershed algorithm with an over-segmentation reduction technique," *arXiv preprint arXiv:1810.03908*, 2018.

[163] D. H. Douglas and T. K. Peucker, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature," *Cartographica: the International Journal for Geographic Information and Geovisualization*, vol. 10, no. 2, pp. 112–122, 1973.

[164] B. Bischke, P. Helber, J. Folz, D. Borth, and A. Dengel, "Multi-task learning for segmentation of building footprints with deep neural networks," in *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2019, pp. 1480–1484.

[165] V. Mnih, *Machine Learning for Aerial Image Labeling*. University of Toronto (Canada), 2013.

[166] E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez, "Can semantic labeling methods generalize to any city? the inria aerial image labeling benchmark," in *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. IEEE, 2017, pp. 3226–3229.

[167] S. Ji, S. Wei, and M. Lu, "Fully convolutional networks for multisource building extraction from an open aerial and satellite imagery data set," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 1, pp. 574–586, 2018.

[168] A. Dutta and A. Zisserman, "The via annotation software for images, audio and video," in *Proceedings of the 27th ACM International Conference on Multimedia*, 2019, pp. 2276–2279.

[169] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the Thirteenth International Con-*

ference on Artificial Intelligence and Statistics. JMLR Workshop and Conference Proceedings, 2010, pp. 249–256.

[170] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440.

[171] R. Cao, J. Zhu, W. Tu, Q. Li, J. Cao, B. Liu, Q. Zhang, and G. Qiu, "Integrating aerial and street view images for urban land use classification," *Remote Sensing*, vol. 10, no. 10, p. 1553, 2018.

[172] N. Audebert, B. Le Saux, and S. Lefèvre, "Joint learning from earth observation and openstreetmap data to get faster better semantic maps," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 67–75.

[173] N. Audebert, B. Le Saux, and S. Lefèvre, "Beyond rgb: Very high resolution urban remote sensing with multimodal deep networks," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 140, pp. 20–32, 2018.

[174] S. Piramanayagam, E. Saber, W. Schwartzkopf, and F. W. Koehler, "Supervised classification of multisensor remotely sensed images using a deep learning framework," *Remote Sensing*, vol. 10, no. 9, p. 1429, 2018.

[175] B. Parajuli, P. Kumar, T. Mukherjee, E. Pasiliao, and S. Jambawalikar, "Fusion of aerial lidar and images for road segmentation with deep cnn," in *Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2018, pp. 548–551.

[176] S. K. Devalla, P. K. Renukanand, B. K. Sreedhar, G. Subramanian, L. Zhang, S. Perera, J.-M. Mari, K. S. Chin, T. A. Tun, N. G. Strouthidis *et al.*, "Drunet: a dilated-residual u-net deep learning network to segment optic nerve head

tissues in optical coherence tomography images," *Biomedical Optics Express*, vol. 9, no. 7, pp. 3244–3265, 2018.

[177] M. A. Islam, S. Naha, M. Rochan, N. Bruce, and Y. Wang, "Label refinement network for coarse-to-fine semantic segmentation," *arXiv preprint arXiv:1703.00551*, 2017.

[178] Z. Wang and S. Ji, "Smoothed dilated convolutions for improved dense prediction," *Data Mining and Knowledge Discovery*, vol. 35, no. 4, pp. 1470–1496, 2021.

[179] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *arXiv preprint arXiv:1511.07122*, 2015.

[180] T. Y. Zhang and C. Y. Suen, "A fast parallel algorithm for thinning digital patterns," *Communications of the ACM*, vol. 27, no. 3, pp. 236–239, 1984.

[181] J. Nunez-Iglesias, A. J. Blanch, O. Looker, M. W. Dixon, and L. Tilley, "A new python library to analyse skeleton images confirms malaria parasite remodelling of the red blood cell membrane skeleton," *PeerJ*, vol. 6, p. e4312, 2018.

[182] A. Chaurasia and E. Culurciello, "Linknet: Exploiting encoder representations for efficient semantic segmentation," in *2017 IEEE Visual Communications and Image Processing (VCIP)*. IEEE, 2017, pp. 1–4.

[183] C. Hazirbas, L. Ma, C. Domokos, and D. Cremers, "Fusenet: Incorporating depth into semantic segmentation via fusion-based cnn architecture," in *Asian Conference on Computer Vision*. Springer, 2016, pp. 213–228.